LEARNING REPRESENTATIONS TOWARD THE UNDERSTANDING OF OUT-OF-DISTRIBUTION FOR NEURAL NETWORKS

A Dissertation
Presented to
The Academic Faculty

By

Gukyeong Kwon

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

August 2021

LEARNING REPRESENTATIONS TOWARD THE UNDERSTANDING OF OUT-OF-DISTRIBUTION FOR NEURAL NETWORKS

Thesis committee:

Dr. Ghassan AlRegib
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Mark A. Davenport
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Justin Romberg
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Eva Dyer
Coulter Department of Biomedical Engineering
Georgia Institute of Technology

Dr. Avinash Ravichandran
AWS AI
Amazon

Date approved: July 14, 2021

I dedicate this thesis to

My Father Yongsoo,

My Mother Sunhee,

My Brother Guyoung.

ACKNOWLEDGMENTS

I am truly grateful to my advisor, Prof. Ghassan AlRegib, for giving me an opportunity to pursue Ph.D. under his guidance. His continuous support, teaching, and trust made it possible for me to overcome challenges and become qualified for Ph.D.. I am certain that his advising not only prepared me for the degree but also impacted my life to become a better researcher and person. I would like to thank Dr. Justin Romberg, Dr. Mark Davenport, Dr. Eva Dyer, and Dr. Avinash Ravichandran for serving on my thesis committee. They shared their valuable time and comments to improve this thesis and their feedback inspired me to make progress on a significant portion of my Ph.D. works.

This thesis could have not been finished without my lab members' help. I would like to thank all former and current members of OLIVES lab. Special thanks goes to my dear friends, Dogancan Temel, Mohit Prabhushankar, and Jinsol Lee. I was very fortunate to have these friends during my Ph.D. journey. With them, I was able to smile, laugh, and enjoy during this journey even with tough struggles. I firmly believe the friendship with them is as precious outcome as my degree from this journey. I am also thankful to our current lab members, Charlie Lehman, Chen Zhou, Joseph Aribido, Yash-yee Logan, Ahmad Mustafa, Ryan Benkert, and Kiran Kokilepersaud. My former lab memebers, Mohammed Aabed, Yazeed Alaudah, Motaz Al-Farraj, Chih-Yao Mao, Zhen Wang, Min-Hun Cheng, Yuting Hu, Amir Shafiq, Tariq Alshawi were amazing mentors who taught me how to conduct research. I am grateful to them for their mentorship.

I would like to thank my family and friends for their support and encouragement. My everlasting gratitude goes to my mother and father. Without them, I could have not been where I am now. Their unconditional love and endless trust were the biggest support for me. No matter what my decision is, they always trusted me and scarified to make my dream come true. I am also grateful to my younger brother for always giving me thoughtful advice and guiding me through difficult decisions. I thank my love, Geuna Ji, for always bringing

me joys and happiness to my life. This Ph.D. journey could have not been possible without her. I am grateful to my dear friends, Mihee Ji and Young Seuk Kim, for their continuous support and thoughtful help. They became my first friends when I came to Atlanta by myself and still remain as my strong supporters and beloved friends. I would like to express my gratitude to my other dear friends, Jiwon Yeon, Hyeonki Jeong, and Hoon Jeong for their support and encouragement.

As I finish up my Ph.D. journey, I am fortunate that I can list these many people around me in my thesis. Their love and friendship will never be forgotten. I sincerely thank them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Data-driven representations achieve powerful generalization performance in diverse information processing tasks. However, the generalization is often limited to test data from the same distribution as training data (in-distribution (ID)). In addition, the neural networks often make overconfident and incorrect predictions for data outside training distribution, called out-of-distribution (OOD). In this dissertation, we develop representations that can characterize OOD for the neural networks and utilize the characterization to efficiently generalize to OOD.

We categorize the data-driven representations based on information flow in neural networks and develop novel gradient-based representations. In particular, we utilize the back-propagated gradients to represent what the neural networks has not learned in the data. The capability of gradient-based representations for OOD characterization is comprehensively analyzed in comparison with standard activation-based representations. We also develop a regularization technique for the gradient-based representations to better characterize OOD. We develop an anomaly detection algorithm named **GradCon** using the gradient constraint and achieve state-of-the-art performance in OOD detection.

We also propose activation-based representations learned with auxiliary information to efficiently generalize to data from OOD. We use an unsupervised learning framework to learn the aligned representations of visual and attribute data. This aligned representation are utilized to calibrate the overconfident prediction toward ID classes. The generalization performance of the aligned representations is validated in the application of generalized zero-shot learning. Our developed GZSL method, **GatingAE**, achieves state-of-the-art performance in generalizing to OOD without using labeled OOD data. Also, balanced performance for both ID and OOD is achieved by mitigating the prediction bias presented in the network. Finally, GatingAE requires significantly less number of model parameters compared to other state-of-the-art methods.

CHAPTER 1
INTRODUCTION

Representation of data is a key element for the success of information processing tasks [1]. As the complexity of data increases, it becomes more challenging to perform target tasks directly using raw data. Representation of data focuses on highlighting specific characteristics of data which is essential for the target task and enables algorithms to easily solve the task. Obtaining effective representations become particularly challenging when the target task need to be achieved for a large amount of data. Since different data possesses various features for the target task, consistently capturing them for a large amount of data in the representations is not a trivial task. This challenge mainly leads to the development of data-driven representations.

Data-driven representations from neural networks contribute to achieve generalizable performance in diverse complicated tasks. Compared to traditional handcrafted representations, data-driven representations are learned from training data and effectively characterize a large scope of unseen test data. However, the data-driven representation still has a limitation. Although powerful generalization performance is achieved through representation learning, the generalization is often limited to test data from the same distribution as training data (in-distribution (ID)). For instance, as shown in Figure 1.1, representations from neural networks trained for `no parking` traffic sign recognition can characterize diverse types of `no parking` signs. However, it cannot capture data from outside the training distribution (out-of-distribution (OOD)) such as other classes of traffic signs or `no parking` signs capture under different conditions. Considering that data in real world scenarios are mostly from OOD, it is important to develop an effective way to learn representations for OOD based on ID data.

The first step to learn representations for OOD is to distinguish between ID and OOD

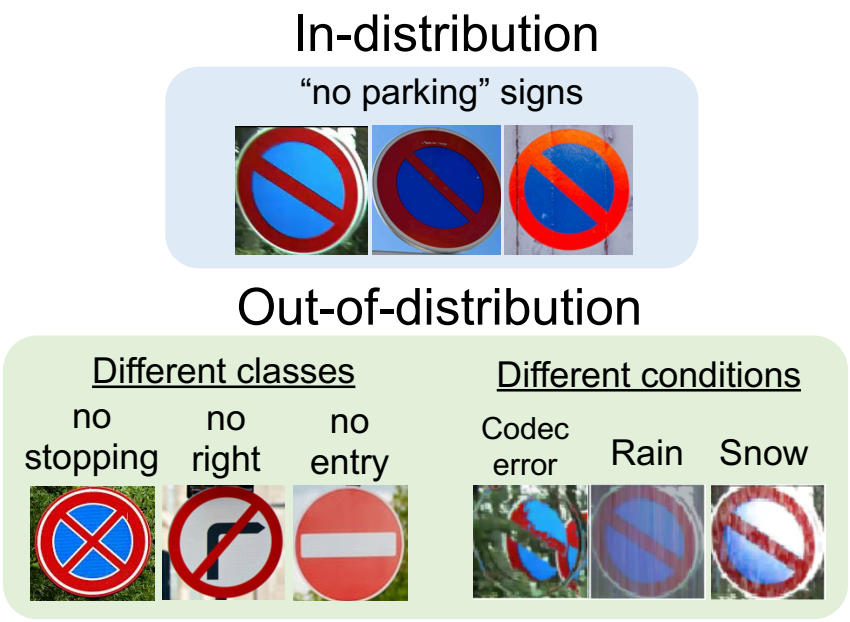


Figure 1.1: Examples of out-of-distribution data.

from the perspective of neural networks. Deep neural networks often fail at indicating when they are likely to make incorrect predictions [2]. Since representations are focused on capturing features presented in ID data, what neural networks do not know in OOD are not thoroughly characterized. As shown in the top of Figure 1.2, assume that the network is trained with only `no parking` traffic signs and a new class of `no stopping` sign is given to the network during testing. The network can easily misclassify `no stopping` as `no parking` by capturing ID features such as the red circle and the blue background in `no stopping`. However, the representations that distinguish between ID and OOD can capture what the network has not learned, which is a red diagonal line from top right to bottom left, and enable to detect the OOD class of `no stopping`. Detection of OOD can be used by the neural network to find a better solution of the target task in OOD. To make the correct prediction in OOD, the neural network still need to learn classes or features in OOD.

Learning OOD can be data-efficiently achieved by characterizing the association between ID and OOD. When the neural network is trained with a sufficiently large amount of

data from ID, features presented in OOD data are not completely new to the network. In this case, OOD data can be characterized by representations that compose different features already learned from ID. An example is illustrated in the bottom of Figure 1.2. Assume that a traffic sign image classifier is trained for `no parking` class and `no right` class in ID. A `no stopping` sign is given to the network during testing. Although the `no stopping` sign is a new class in OOD to the network, it can be still represented by associating `no parking` features and the red diagonal like from top right to bottom left presented in `no right`. To guide this association, we only need auxiliary information such as textual description, "`no stopping` is `no parking` with an additional red cross line from top right to bottom left" rather than a large amount of data for `no stopping` signs. This example conceptually highlights that the neural networks can generalize to OOD without expensive data collection and re-training, which is a practical solution to handle a broad range of OOD in real world scenarios.

In this dissertation, I focus on information flow in neural networks to develop representations for detecting and generalizing to OOD. In particular, both activation from forward propagation and gradients from backpropagation are utilized as key components to construct the representations. First, I develop gradient-based representations which can clearly separate OOD from ID. Gradients are generated through backpropagation to train neural networks by minimizing designed loss functions [3]. During training, the gradients with respect to the weights provide directional information to update the neural network and learn knowledge that it has not learned. Since the neural network has been already trained with ID data, the gradients from ID data do not guide a significant change of the current weight. However, the gradients from OOD data guide more drastic updates on the network to fully represent data. Therefore, the gradients can be used to distinguish OOD from ID by measuring *how much model update is required by the input compared to ID data*.

The advantages and the limitations of gradient-based representations are also thoroughly analyzed in comparison with the activation-based representations. While gradient-

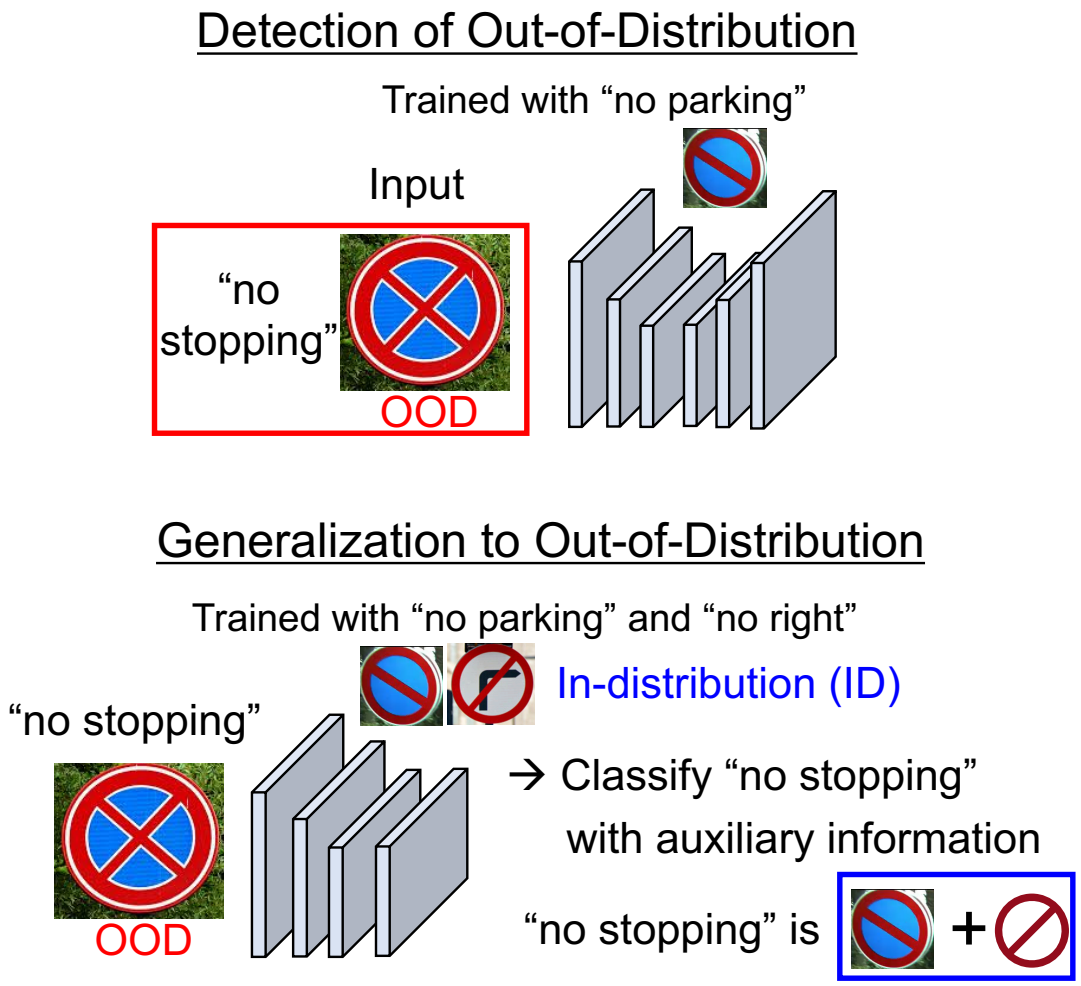


Figure 1.2: Detection of out-of-distribution (OOD) and Generalization to OOD using in-distribution (ID) features.

based representations can distinguish between ID and OOD, they are limited to represent what the network does not know. Therefore, it is not feasible to perform target tasks such as classification only using gradient-based representations. On the other hand, the activation-based representations are the most commonly used form of representations which contain rich information about training data to perform target tasks. Since the activation-based representations characterize what the network knows, they suffer in detecting OOD compared to the gradient-based representations. Based on these advantages and limitations of the activation- and the gradient-based representations, we seek for complementing each other for detecting and generalizing to OOD.

Finally, the activation-based representations learned with auxiliary data is developed for generalization to OOD. In particular, we measure the alignment between the query data and auxiliary information for ID and OOD to detect OOD. The alignment with the auxiliary information for OOD characterizes what the network does not know. In addition, the neural network use the auxiliary information to learn the association between ID and OOD and to generalize to OOD. One of the challenges in learning with the auxiliary information is that the representations are biased toward ID for the target task. Since only auxiliary information is provided for OOD while fully annotated data is available for ID, the neural networks easily overfit to ID and the representations are optimized to perform well only for ID. This bias should be calibrated to make the neural networks generalize to not only ID but also to OOD. To overcome this challenge, we first detect OOD and the detection results are utilized to calibrate the prediction.

Developed representations for OOD contribute to a broad range of machine learning applications where only limited data is available for learning. Expensive data collection and annotation often limit the size of training data, which results in more OOD data for the network. Developed representations characterize diverse OOD data and their effectiveness is rigorously validated in the applications including anomaly detection and generalized zero-shot learning. We show that these representations can be utilized to achieve the generalized performance of learning algorithms while effectively utilizing available limited annotated data.

CHAPTER 2
LITERATURE SURVEY

In this chapter, we comprehensively review the literature related to representation learning for OOD. I first explain the development of representation learning techniques and introduce existing approaches for OOD detection. In particular, we focus on techniques to obtain the representations that distinguish between ID and OOD. We also review developed representations for the generalization of neural networks to OOD. Learning with auxiliary information and bias calibration techniques will be mainly explained.

2.1 Representation Learning

Representation is obtained through the transformation of data to make target tasks easier for the algorithms. Therefore, the representation focuses on highlighting important features of data which can be useful to solve the task. Since the useful features of data vary depending on the task, the representation is also task-dependant. For instance, the representation is desired to minimize the intra-class distance and maximize the inter-class distance between data for classification tasks to be easily performed. However, in dictionary learning, sparse representation is desired to achieve the robustness and the interpretability of features [4, 5]. As availability of data increases, target tasks become more complicated and more descriptive representations are required. Hence, techniques to obtain representations from handcrafting to data-driven approaches have advanced to achieve the success in complicated information processing tasks.

Traditional signal processing techniques have focused on deriving handcrafted representations. Frequency domain representations obtained by techniques including but not limited to Fourier transform, Wavelet transform, and Curvelet transform are one of the most traditional representations which characterize diverse frequency components presented in

data. These representations become particularly useful for applications which require adaptive processing for different frequency bands such as data compression or denoising [6, 7]. Image processing techniques have also focused on directly extracting specific visual feature such as edges [8], color [9], and structures [10]. These representations for visual features are utilized as fundamental components for diverse computer vision and image processing tasks such as image, video quality assessment and super resolution [11]. Since handcrafted representations are obtained from closed-form expressions for feature engineering, representations are interpretable and easy to manipulate. However, as data that the task needs to be performed on becomes larger, the feature engineering faces challenges in generalizing and the representation capability becomes limited.

Representation learning from a large amount of data enables to obtain generalizable representations. Depending on utilized supervision, data-driven representations can be categorized into supervised, unsupervised, semi-supervised, and self-supervised representations. First, supervised learning aims at training neural networks which accurately predict data labels such as class, bounding boxes, and segmentation maps. Representations are learned to discriminate different labels and widely utilized for tasks such as object recognition, detection, and segmentation [12, 13, 14]. Although supervised representations achieve successful performance in diverse large-scale datasets, they are limited by expensive data collection and annotation. This limitation becomes particularly problematic for applications where domain expertise is required to annotate the data such as seismic or medical data interpretation. To overcome this limitation, representation learning techniques which do not use fully annotated data are developed.

Unsupervised learning does not require annotated data while constraints are imposed during training to generate representations with desired characteristics. Autoencoders and generative models fall into this category. An autoencoder learns a regularized latent representation of input to reconstruct it as output [15]. Depending on constraints imposed in the latent representation, variants of the autoencoder exist. A sparse autoencoder is trained

with l_1 regularization in the latent space and enforces the latent variables to be sparse [16]. This constraint encourages the network to learn the interpretable structure of data in the latent space. A variational autoencoder (VAE) is another variant which constrains the latent space to follow the Gaussian distribution [17]. The Gaussian constraint enables to generate images by sampling latent variables from Gaussian distribution and feeding them into the decoder. Generative adversarial network (GAN) implicitly learns the probability distribution of input data and generate new data similar to input data [18]. GANs show disentangled representations which characterize visual concepts independently in each dimension of latent space can be learned through adversarial training [19]. Although unsupervised learning overcomes the limitation of expensive annotation, the neural networks cannot get explicit supervision during training, which leads to performance degradation compared to supervised networks.

Semi-supervised learning is a hybrid between supervised and unsupervised learning since it learns representations with both labeled and unlabeled data. Self-training is one of the most widely used semi-supervised learning techniques [20]. Supervised models are first trained with the labeled data and used to predict pseudo-labels for unlabeled data. Unlabeled data whose pseudo-labels are predicted with high confidence is included for the re-training of the supervised models [21]. Co-training extends self-training and uses multiple supervised classifiers to generate more accurate pseudo-labels [22]. For co-training, it is essential to prevent classifiers from being excessively correlated so that they can complement each other. Several perturbation-based models are proposed based on the intuition that model prediction should be consistent for similar data [23, 24]. Hence, each data with and without noise are evaluated by the model and consistency costs are applied as regularization during training. This consistency constraint enables to accurately propagate correct labels for unlabeled data. Semi-supervised learning enables to avoid expensive manual annotation but unlabelled data should be carefully chosen to complement labeled data to obtain useful representations.

Self-supervised learning aims and learning representations without relying on manual supervision. In comparison with unsupervised learning, self-supervised representations are often learned by solving a pretext task the goal of which is to predict the properties of transformation applied on data. For instance, visual representations can be learned by solving image Jigsaw puzzles, predict the angle of image rotation, image inpainting, and colorization [25, 26, 27, 28]. Also, based on the motivation that solving pretext tasks of predicting properties of transformation makes representations covariant to the transformations, several method propose pretext invariant representations [29, 30]. These approaches encourage the representations of original data and transformed data to be similar. The authors show these representations are more effective than those from supervised pretraining on detection and segmentation tasks.

In this dissertation, we focus on learning paradigms suitable to obtain representations of OOD. Based on the assumption that we only have access to ID data, we explore unsupervised and supervised representation learning frameworks for both scenarios where we have ID data with and without annotations. In particular, we first develop an unsupervised learning framework to detect OOD and show representations for detecting OOD can be successfully incorporated in a supervised learning framework for the generalization to OOD.

2.2 Detection of Out-of-distribution

We establish a new categorization of representations based on the information flow that the representations are obtained from. Most of representations from neural networks are obtained in a form of activation, which is the outcome from the feedforward propagation of data through the networks. We refer to them as activation-based representations. On the other hand, we develop novel representations which are constructed by backpropagated gradients, called gradient-based representations. While various techniques for learning activation-based representations are developed, the representation capability of gradient-

based representations remains largely unexplored. We explain literature related to activation- and gradient-based representations in the context of OOD detection.

Research on OOD detection has been conducted under the several names of topics including OOD detection, anomaly detection, novelty detection, and one-class classification. A common goal of these research topics is to learn representations that differentiate OOD from ID when only ID data is available during training. However, different types of OOD have been utilized to validate the effectiveness of the representations. First, datasets different from the training dataset form one type of OOD. For example, a general scene image dataset can be a OOD for the neural network trained with a handwritten digit image dataset. Second, different classes of data not used for training can be also considered as OOD. When the neural network is trained with digit 0 images in the handwritten digit image dataset, remaining digit images in the same dataset are in OOD. Finally, perturbations applied on data shift the data distribution and make data OOD. The distribution of digit images with distortions or artifacts is OOD for the distribution of pristine digit images. We broadly review existing representation learning techniques dealing with different types of OOD in following subsections.

2.2.1 Activation-based Representations

Confidence-calibrated activation-based representations are explored for OOD detection. Prediction confidence is often obtained from the output of the softmax classifier. When prediction confidence is ideally calibrated, the neural networks predicts labels for OOD data with less confidence and we can differentiate OOD from ID using the confidence value. However, the authors in [2] point out that class prediction confidence is poorly calibrated in modern deep neural networks. The deep neural networks confidently predicts labels for OOD data, which makes it challenging to distinguish OOD from ID. To remedy this problem, they show temperature scaling which simply scales and softens the softmax output distribution is effective for confidence calibration [31, 32]. [33] also use gradient-

based input preprocessing and temperature scaling and analyze the contribution of them for OOD detection. In [34], Mahalanobis distance-based confidence score is proposed and it shows better performance in OOD detection compared to softmax-based confidence score. [35] uses a neural network which generates not only prediction logits but also a confidence logit to learn calibrate confidence scores for data. In particular, the task loss is designed to be minimized when the confidence score is correctly estimated. The confidence score is directly used to detect OOD. The authors in [36] note that existing works [33, 34, 37] have limitations that they require OOD data for hyperparameter tuning or regularization. [36] overcomes this limitation by using decomposed confidence scores and a modified input preprocessing method which uses ID data to tune the parameter. While learning confidence-calibrated activation-based representations is an intuitive solution for OOD detection, most of the methods require input preprocessing or regularization which may sacrifice the representation capability for the original target task.

Several works propose learning constrained activation-based representations for the detection of OOD. In particular, when the representations are constrained during training with ID data, OOD data results in representations that deviate from the constraint while ID generates relatively more constrained representations. By measuring the deviation from the constraint imposed on representations, ID and OOD can be separated. Learning to constrain encoded representations inside hyperplane or hypersphere is actively explored to detect anomalies from OOD. One-Class support vector machine (OC-SVM) learns a maximum margin hyperplane which separates data from the origin in the feature space [38]. Abnormal data is expected to lie on the other side of normal data and separated by the hyperplane. The authors in [39] extend the idea of OC-SVM and propose to learn a smallest hypersphere that encloses the most of training data in the feature space. In [40], a deep neural network is trained to constrain the activation-based representations of data into the minimum volume of hypersphere. For a given test sample, an anomaly score is defined by the distance between the sample and the center of hypersphere.

Learning the constrained activation-based representations from autoencoders has been another dominant approach for OOD detection. The autoencoder generates two well-constrained representations, which are latent representation and reconstructed data representation through unsupervised learning. Based on these constrained representations, latent loss or reconstruction error have been widely used as anomaly scores which characterize the probability of the data being from OOD. In [41], [42], the authors argue that anomalies cannot be accurately projected in the latent space and are poorly reconstructed. Therefore, they propose to use the reconstruction error to detect anomalies. The authors in [43] fit Gaussian mixture models (GMM) to reconstruction error features and latent variables and estimate the likelihood of inputs to detect anomalies. The neural network estimates the parameters of GMM to predict the mixture membership of query data. In [44], the authors develop an autoregressive density estimation model to learn the probability distribution of the latent representation. The autoregressive estimator consists of masked fully connections and masked stacked convolution. The likelihood of the latent representation and the reconstruction error are used to detect abnormal OOD data.

Adversarial training is also utilized to differentiate the representation of abnormal OOD data. In general, a generator learns to generate realistic data similar to training data and a discriminator is trained to discriminate whether the data is generated from the generator (fake) or from training data (real) [18]. The discriminator learns a decision boundary around training data and is utilized as an abnormality detector during testing. In [45], the authors adversarially train a discriminator with an autoencoder to classify reconstructed images from original images and distorted images. In this case, the distorted images are utilized to model OOD and the discriminator is utilized as an anomaly detector during testing. In [46], the mapping from a query image to a latent variable in a generative adversarial network (GAN) [18] is estimated. The loss which measures visual similarity and feature matching for the mapping is utilized as an anomaly score. The authors in [47] use an adversarial autoencoder [48] to learn the parameterized manifold in the latent space and estimate

probability distributions for anomaly detection. In [49], a GAN is trained to generate OOD samples that result in uniform distribution of the classifier output. When the classifier is trained with ID, uniform distribution of the classifier output indicates that the classifier is not confident in predicting classes of the generated samples. The classifier is jointly trained with the GAN and make confidence-calibrated predictions to detect OOD.

Aforementioned works exclusively focus on distinguishing between ID and OOD using the activation-based representations. In particular, most of the algorithms use adversarial networks or likelihood estimation networks to further constrain activation-based representations. These networks often require a large amount of training parameters and computations. We show that a directional constraint imposed on the gradient-based representations enables to achieve the state-of-the-art anomaly detection performance using only a backbone autoencoder with significantly less number of model parameters.

2.2.2 Gradient-based Representations

Backpropagated gradients as data representations are explored in this dissertation. The backpropagated gradients have been utilized in diverse applications including but not limited to visualization, adversarial attacks, and image classification. The backpropagated gradients have been widely used for the visualization of deep networks. In [50], [51], information that networks have learned for a specific target class is mapped back to the pixel space through the backpropagation and visualized. The authors in [52] utilize the gradients with respect to the activation to weight the activation and visualize the reasoning for prediction that neural networks have made. The authors in [53] use gradients to generate contrastive explanations in neural networks. When the neural network predicts a class P for a given image, a class of interest Q is given as a label to compute the loss and generate gradients. These gradients provide explanations why the neural network predicts P rather than Q. An adversarial attack is another application of gradients. In [54], [55], the authors show that adversarial attacks can be generated by adding an imperceptibly small vector

which is the signum of input gradients. Adding small perturbations based on gradients is also utilized as an input processing technique for OOD detection [33, 34]. The gradient of the softmax score with respect to input is computed and the perturbation is added on input to increase the softmax score of any given input. This perturbation has stronger effect on ID data and separates softmax scores from ID and OOD. Several works have incorporated gradients with respect to the input in a form of regularization during the training of neural networks to improve the robustness [56], [57], [58]. Although existing works have shown that the gradients with respect to the input or the activation can be useful for diverse applications, the gradients with respect to the weights of neural networks have not been actively explored aside from its role in training deep networks.

A few works have explored the gradients with respect to the model parameters as features for data. The authors in [59] propose to use Fisher kernels which are based on the normalized gradient vectors of the generative model for image categorization. The authors in [60, 61] characterize information encoded in the neural network and utilize Fisher information to represent tasks. In [62], the gradients of the neural network are utilized to classify distorted images and objectively estimate the quality of them. [63] extracts gradients from each layer of a supervised image classifier and use them as features to perform OOD detection. In particular, a confounding label is used to generate the gradients and a OOD detector is trained with the gradients from ID and OOD. The gradients have been also studied as a local liner approximation to a neural network [64]. Our approach differs from other existing works in two main aspects. First, we generalize the Fisher kernel principal using the backpropagated gradients from the neural networks. Since we use the backpropagated gradients to estimate the Fisher score of normal data distribution, the data does not need to be modeled by known probabilistic distributions such as a GMM. Second, we use the gradients to represent information that the networks have not learned. In particular, we provide our interpretation of gradients which characterize abnormal information for the neural networks and validate their effectiveness in OOD detection.

2.3 Generalization to Out-of-Distribution

We develop activation-based representations with auxiliary data for the generalization to OOD. There have been a number works focusing on learning activation-based representations with auxiliary information but in this literature survey, we specifically focus on those representations for generalizing learned knowledge to OOD. Since we do not have access to OOD data, the auxiliary information should contain meaningful high-level concepts that transcend classes in both ID and OOD [65]. For example, assume that "bee" or "cat" are ID classes while "zebra" is a OOD class. The auxiliary information of their common attribute "stripe" will allow neural networks to utilize stripe visual features learned from ID class to characterize the OOD "zebra" class. Hence, it is critical to learn strong association between data and auxiliary information for generalization to OOD. We review various techniques to learn aligned representations for both visual data and auxiliary information.

We also explain bias calibration for activation-based representations learned with auxiliary information. While generalizing to OOD, we expect neural networks to successfully perform target tasks for both ID and OOD. However, when test data is from both ID and OOD, the neural networks are easily biased to perform well for ID data while poorly for OOD. This is because we do not use any OOD data to learn representations while fully annotated data is used to learn ID representations. To overcome this challenge, several techniques have been developed to calibrate the prediction of neural networks. We review details of these works and highlight the novelty in our approach.

2.3.1 Aligned Representations for Visual Features and Attributes

Learning aligned representations for visual data and auxiliary information of attribute data is widely explored for the generalization to OOD. In [66], semantic knowledge learned from text data is used as a type of attributes and aligned with visual representations in the joint embedding space. In particular, a skip-gram text model is trained to predict adjacent

terms using a corpus from wikipedia. This text model is used to obtain features for the text label. The distance between image features from a image classification model and the corresponding text label features are minimized in the embedding space. The authors in [67] use autoencoders to obtain representations for visual data and textual attributes. Also, a cross-modality distribution matching constraint which minimizes the maximum mean discrepancy between visual and attribute features is imposed to align the representations. In [68], the joint representation is learned by matching the graph structure of semantic space and model space. The semantic space is defined by representations of attributes and the model space is constructed by using classes as nodes and model weights as edges. The coordinates in the model space is considered as the projection of vertices in the semantic space to align both graphs. The authors in [69] propose to map images to the semantic embedding space through the convex combination of semantic embedding vectors. A latent probabilistic model is proposed in [70] to learn the statistical relationship between visual and attribute representations. To be specific, a binary predictor is trained to output a likelihood score which indicates whether visual data and attributes are corresponding or not. The authors in [71, 72] propose to learn compatibility functions that can relate the visual features with attribute representations. The compatibility functions are learned by modelling the relationship between features, attributes, and classes in a linear two-layer network. The first layer related features to attributes, and the second layer is fixed as given relationship between attributes and classes. In [73], the authors propose to learn the joint representations through contrastive learning and generalize the representations to unseen classes by imposing a transferability constraint. In [74], a dense attribute-based attention mechanism is proposed to align attributes with local visual features instead of global feature vectors from images. In this dissertation, we use a two-stream autoencoder which shares representations from each stream to learn joint embedding.

2.3.2 Generative Models for Feature Generation

Generative models such as generative adversarial networks (GANs) [18] and variational autoencoders (VAEs) [17] have been widely used to generate OOD data from attributes. Generated OOD can be utilized to learn representations for OOD. Compared to the aligned representations of visual data and attributes in a shared embedding space, generative models can be considered as mapping from attributes to visual data. The authors in [75] use a Wasserstein GAN (WGAN) [76] conditioned on attribute information to generate unseen visual features. To ensure the generated features to be class discriminant, classification loss over generated features are also imposed during training. [77] uses a multi-modal cycle consistency loss which enforces accurate reconstruction from generated visual features to attributes. By doing so, the authors argue that GANs generated visual features that share more semantic concepts with attributes. [78] proposes gradient matching loss on the WGAN to generate class discriminant visual features of OOD. In particular, the authors hypothesize that when a generative model learns true class manifolds, the gradients from generated and real samples should be highly correlated. They show that generated OOD enable to learn effective representations for OOD classification. [79] uses a diffusion regularization which aims at reducing the reluctant dimensions in the synthesized data and diffusing information to all the dimensions. In [80], a modified WGAN is used to generate visual prototypes in an episode-based training setup. In each episode during training, ID data is split into a support set and a refining set. The neural networks learn base representations from ID data and attributes. By performing a target task of OOD classification on the refining set and update the model, the model accumulates the experience of representing OOD data. [81] uses a single conditional generator trained via an alternating backpropagation algorithm to generate visual features. Instead of GANs, several methods are based on conditional VAEs to synthesize samples [82, 83]. In [84], a two-stream VAE is utilized to generate latent representations for OOD and the latent representations are used to train a classifier. In comparison with the standard VAEs, cross-alignment loss

is imposed on reconstruction space and distribution-alignment loss is used for the latent representations. Both losses aim at learning aligned representations for ID data and corresponding attributes. This alignment enables to generate OOD features by using attributes of OOD data. Although generative model-based approaches have achieved successful performance in GZSL, generative models often require a large number of model parameters. Our developed method is not relying on any generative models and requires significantly less computational resources to train the model.

2.3.3 Calibration of Biased Prediction Toward Seen Classes

Several works have focused on preventing models from making a biased prediction toward ID data when test data is drawn from both ID and OOD. [85] proposes a gating model which estimates the local outlier probability for unseen class detection. In particular, each in-distribution class is modeled by Gaussian distribution the mean and the variance of which are semantic word vector and the covariance of ID classes. A simple threshold on the estimated likelihood is used as an OOD detector. The authors in [86] propose diverse ways to detect OOD data and use them for OOD classification. First, a ID classifier is trained to classify ID data and an OOD classifier is trained with generated data from a GAN. In addition, the authors train another classifier to select either the ID or OOD classifier to use for the query data. This model selector can be considered as a hard-gating model. A soft-gating model is also proposed to combine predictions for the ID and the OOD classifier to predict the final class of the query test data. Finally, they add a classifier trained on both ID and OOD data to correct predictions from the separate ID and OOD classifiers. In [87], adaptive confidence-based smoothing is utilized with the soft-gating model which combines prediction scores from the seen and the unseen expert. OOD detection is performed based on the top-k output of the softmax classifier. Also, the authors use Laplace smoothing to incorporate the prior information about ID and OOD. [88] filters out seen class samples by thresholding the entropy of the predicted scores and predicts the

seen and the unseen classes separately. In addition auto-searched semantic-visual embedding is developed for unseen OOD image recognition. The authors in [89] propose using temperature scaling [32] and an entropy-based regularization to mitigate the overfitting on ID class data. [90] calibrates the ID class prediction by using calibrated stacking which reduces the prediction scores for ID classes. In this dissertation, we develop representation learning techniques which learns the activation-based representations with auxiliary information both for detecting and generalizing to OOD.

CHAPTER 3

GRADIENT-BASED REPRESENTATIONS FOR OUT-OF-DISTRIBUTION DETECTION

Representations from neural networks plays a key role in anomalous OOD detection. The representations are expected to clearly differentiate normal data from abnormal data. To achieve the separation, most of existing anomaly detection algorithms deploy a representation obtained in a form of activation. The activation-based representation is constrained during training. During inference, deviation of activation from the constrained representation is formulated as an anomaly score. In Figure 3.1, we demonstrate an example of a widely used activation-based representation from an autoencoder. Assume that the autoencoder is trained with digit '0' and learns to accurately reconstruct curved edges. When an abnormal image, digit '5', is given to the network, the top and bottom curved edges are correctly reconstructed but the relatively complicated structure of straight edges in the middle cannot be reconstructed. Reconstruction error measures the difference between the target and the reconstructed image and it can be used to detect anomalies [44], [42]. The reconstructed image, which is the activation-based representation from the autoencoder, characterizes what the network knows about input. Thus, abnormality is characterized by measuring *how much of the input does not correspond to the learned information of the network*.

In this chapter, we propose using gradient-based representations to detect abnormal OOD by characterizing model updates caused by data. Gradients are generated through backpropagation to train neural networks by minimizing designed loss functions [3]. During training, the gradients with respect to the weights provide directional information to update the neural network and learn knowledge that it has not learned. The gradients from normal data do not guide a significant change of the current weight. However, the gradi-

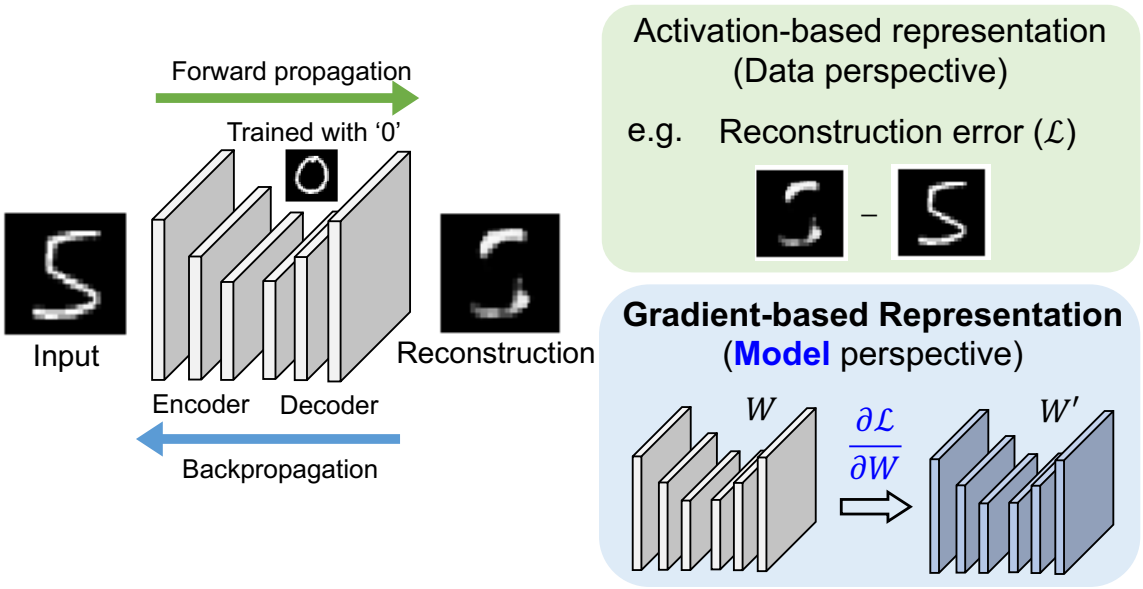


Figure 3.1: Activation and gradient-based representation for anomaly detection. While activation characterizes how much of input correspond to learned information, gradients focus on model updates required by the input.

ents from abnormal data guide more drastic updates on the network to fully represent data. In the example given in Figure 3.1, the autoencoder needs larger updates to accurately reconstruct the abnormal image, digit '5', than the normal image, digit '0'. Therefore, the gradients can be utilized as representations to characterize abnormality of data. We propose to detect anomalies by measuring *how much model update is required by the input compared to normal data*.

The gradient-based representations have several advantages compared to the activation-based representations, particularly for OOD detection. First of all, the gradient-based representations provide abnormality characterization at different levels of data abstraction. The deviation of the activation-based representations from the constraint, often formulated as a loss ($\mathcal{L}$), is measured from the output of specific layers. On the other hand, the gradients with respect to the weights ($\frac{\partial \mathcal{L}}{\partial W}$) can be obtained from any layer through backpropagation. This enables the algorithm to capture fine-grained abnormality both in low-level characteristics such as edge or color and high-level class semantics. In addition, the gradient-based representations provide directional information to characterize anomalies. The loss in the

activation-based representation often measures the distance between representations of normal and abnormal data. However, by utilizing a loss defined in the gradient-based representations, we can use vectors to analyze direction in which the representation of abnormal data deviates from that of normal data. Considering that the gradients are obtained in parallel with the activation, the directional information of the gradients provides complementary features for anomaly detection along with the activation.

The gradients as representations have not been actively explored for OOD detection. The gradients have been utilized in diverse applications such as adversarial attack generation and visualization [50], [54]. However, to the best of our knowledge, our developed gradient-based representation is the first attempt to explore the representation capability of backpropagated gradients for anomalies. We validate the effectiveness of gradient-based representations for OOD detection through comprehensive experiments. In particular, we compare the gradients with the activation-based representations and highlight the effectiveness of gradient features in OOD detection. Also, we perform OOD detection for different classes and conditions of inputs to show the generalizability of gradient features for different types of OOD. The contributions of this chapter are three folds:

i We propose a framework to characterize OOD from the model perspective using gradients.

ii We analyze the representation capability of the gradient compared to activations through controlled experiments.

iii We validate the generalizability of gradient features for different classes and input conditions.

3.1 Geometric Interpretation of Gradients

We use an autoencoder, which is an unsupervised representation learning framework to explain the geometric interpretation of gradients. An autoencoder consists of an encoder,

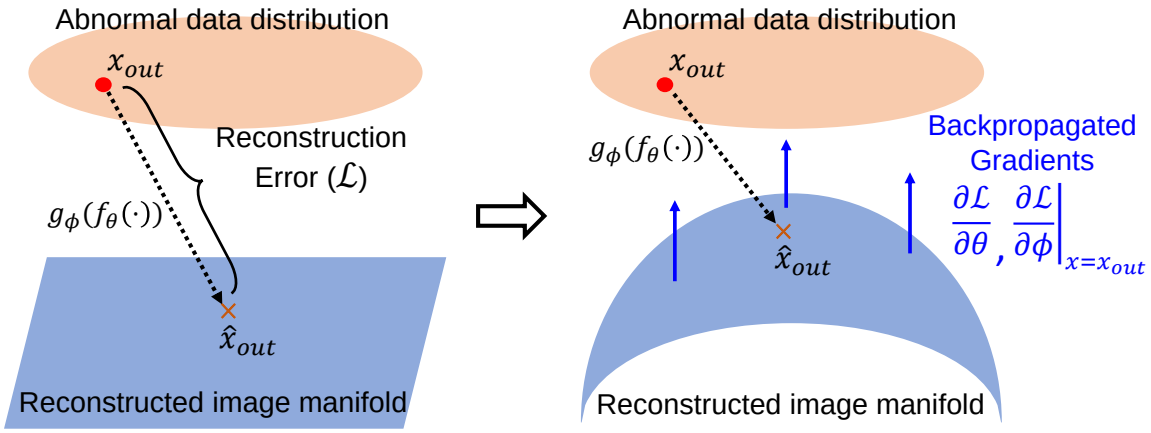


Figure 3.2: Geometric interpretation of gradients.

f_θ, and a decoder, g_ϕ. From an input image, x, a latent variable, z, is generated as $z = f_\theta(x)$ and a reconstructed image is obtained by feeding the latent variable into the decoder, $g_\phi(f_\theta(x))$. The training is performed by minimizing a loss function, $J(x;\theta,\phi)$, defined as follows:

$$J(x;\theta,\phi) = \mathcal{L}(x, g_\phi(f_\theta(x))) + \Omega(z;\theta,\phi), \tag{3.1}$$

where $\mathcal{L}$ is a reconstruction error, which measures the dissimilarity between the input and the reconstructed image and Ω is a regularization term for the latent variable.

We visualize the geometric interpretation of backpropagated gradients in Figure 3.2. The autoencoder is trained to accurately reconstruct training images and the reconstructed training images form a manifold. We assume that the structure of the manifold is a linear plane as shown in the figure for the simplicity of explanation. During testing, any given input to the autoencoder is projected onto the reconstructed image manifold through the projection, $g_\phi(f_\theta(\cdot))$. Ideally, perfect reconstruction is achieved when the reconstructed image manifold includes the input image. Assume that abnormal data distribution is outside of the reconstructed image manifold. When an abnormal image, x_{out}, sampled from the distribution is input to the autoencoder, it will be reconstructed as $\hat{x}_{out}$ through the projection, $g_\phi(f_\theta(x_{out}))$. Since the abnormal image has not been utilized for training, it will be poorly reconstructed. The distance between x_{out} and $\hat{x}_{out}$ is formulated as the reconstruction er-

ror and characterizes the abnormality of the data as shown in the left side of Figure 3.2. The gradients with respect to the weights, $\frac{\partial \mathcal{L}}{\partial \theta}, \frac{\partial \mathcal{L}}{\partial \phi}$, can be calculated through the back-propagation of the reconstruction error. These gradients represent required changes in the reconstructed image manifold to incorporate the abnormal image and reconstruct it accurately as shown in the right side of Figure 3.2. In other words, these gradients characterize orthogonal variations of the abnormal data distribution with respect to the reconstructed image manifold.

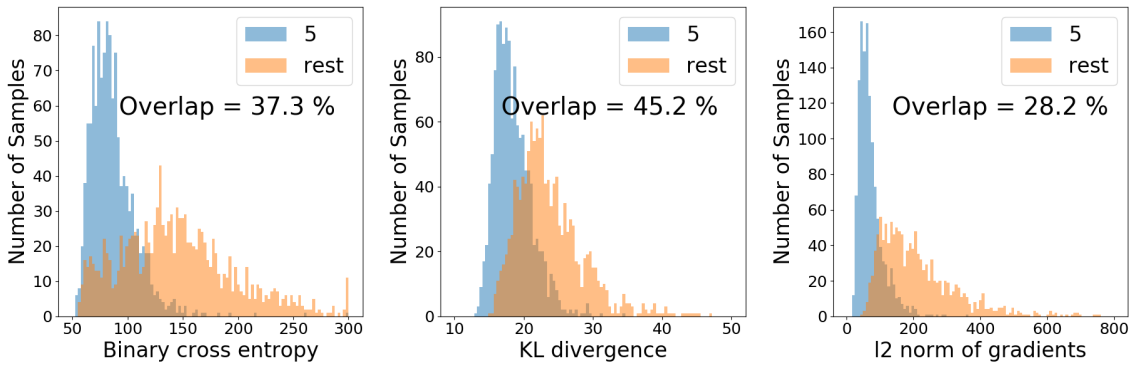
The interpretation of gradients from the data manifold perspective highlights the advantages of gradients in anomaly detection. In activation-based representations, the abnormality is characterized by distance information measured using a designed loss function. On the other hand, the gradients provide directional information, which indicates the movement of manifold in which data representations reside. This movement characterizes, in particular, in which direction the abnormal data distribution deviates from the representations of normal data. Furthermore, the gradients obtained from different layers provide a comprehensive perspective to represent anomalies with respect to the current representations of normal data. Therefore, the directional information from gradients can be utilized as complementary information to the distance information from the activation.

3.2 Statistical Analysis on Gradient-based Representations

We perform statistical analysis on both activation-based and gradients to show the effectiveness of them in characterizing novel data. We train a VAE [17] by minimizing a loss defined as follows:

$$J(x;\theta,\phi) = -\mathbb{E}_{g_\phi(z|x)}[\log f_\theta(x|z)] + \mathrm{KL}[g_\phi(z|x)||f(z)], \tag{3.2}$$

where KL is the Kullback Leibler divergence between two distributions and we assume $f(z) = N(z|\mathbf{0}, I)$. Therefore, KL divergence constrains the latent space of VAE to be the



(a) Reconstruction error (b) Latent loss (c) Gradient

Figure 3.3: Statistical deviation between inliers and outliers.

Gaussian distribution. The first term in the loss corresponds to the reconstruction error, $\mathcal{L}$, and calculated using binary cross entropy. The second term is the latent loss, Ω in Equation 3.1. We train the VAE using digit '5' images in MNIST [91] which are considered as inliers and other digit images are considered as outliers. We obtain the reconstruction error and the latent loss by passing test images through the VAE pre-trained with digit '5'. The gradients are extracted from the first layer of the decoder through the backpropagation of the reconstruction error from each test image.

We visualize histograms of the reconstruction error, the latent loss, and the ℓ_2 norm of gradients in Figure 3.3 (a), (b), (c), respectively. Furthermore, we provide percentages of samples in the overlapped region of the histograms to quantify the separation between two distributions from the inliers and the outliers. Ideally, large separation between the inliers and the outliers is desired for effective OOD detection. As shown in these histograms, the ℓ_2 norm of backpropagated gradients, which measures the magnitude of gradients, better separates the inliers and the outliers than the reconstruction error and the latent loss. This shows that the magnitude of the gradients is more informative in characterizing abnormal data compared to other activation-based measures. In the following section, we utilize both magnitude and direction information of gradients by using them as features for OOD detection and highlight the performance from gradient features.

Figure 3.4: 14 different traffic signs in CURE-TSR.

3.3 Dataset Generation

We create a dataset to rigorously validate the effectiveness of gradient features in detecting diverse types of OOD. Most of existing anomaly detection algorithms are validated for detecting OOD classes. However, in real-world scenarios, there can be more various types of OOD than OOD classes. For instance, environmental challenging conditions, processing artifacts, and distortion can create different OOD for models trained with pristine ID data. This motivates us to create a dataset which contains diverse challenging conditions that can be encountered in a real application such as autonomous driving. Our dataset is named Challenging Unreal and Real Environments for Traffic Sign Recognition (`CURE-TSR`) dataset which is the most comprehensive publicly-available traffic sign recognition dataset with controlled challenging conditions.

We compare CURE-TSR with other existing traffic sign recognition datasets to highlight the advantages of CURE-TSR in terms of its scale and diverse challenging conditions. Timofte *et al.* [92] introduced the Belgium traffic sign classification (BelgiumTSC) dataset whose images were acquired with a van that had 8 roof-mounted cameras. Acquisition vehicle cruised in streets of Belgium and images were captured every meter. A subset of these images were selected and traffic signs were cropped to obtain the BelgiumTSC dataset. Stallkamp *et al.* [93, 94] introduced the German traffic sign recognition benchmark (GTSRB) dataset, which was acquired during daytime in Germany. Each traffic sign

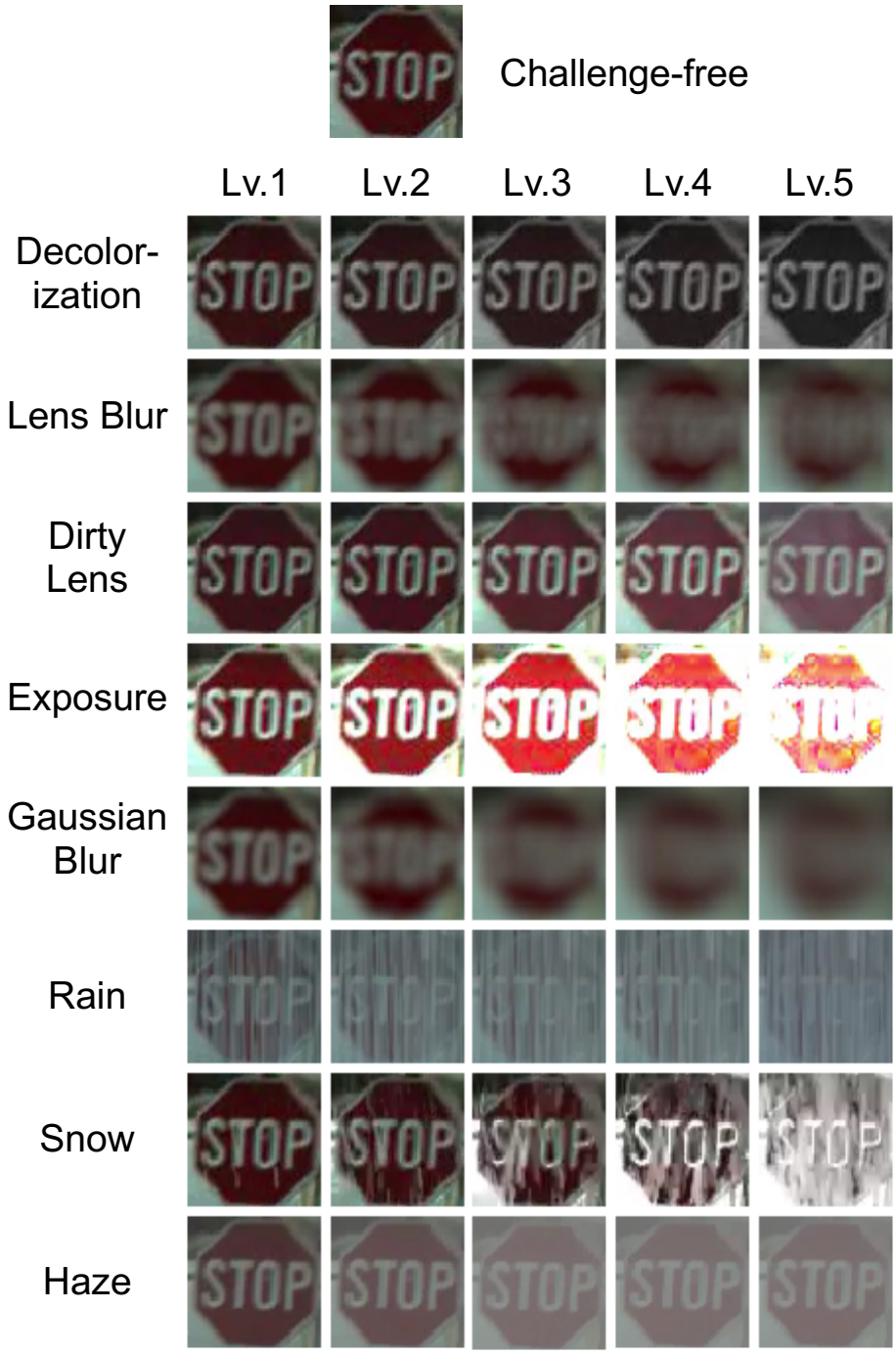


Figure 3.5: A challenge-free stop sign and stop signs with 8 different challenge types and 5 different challenge levels. Challenging conditions become more severe as the level becomes higher.

instance in the dataset is adjusted to have 30 images. BelgiumTSC and GTSRB datasets are limited in terms of challenging environmental conditions and they do not include meta-data related to the type of challenging conditions or their levels. Because of limited control in data acquisition setup, it is not possible to perform controlled experiments with these datasets. The total number of annotated signs including BelgiumTSC and GTSRB datasets is around $60,000$, which may not be sufficient to test the robustness of recognition algorithms comprehensively against OOD. To compensate the shortcomings in the literature, we introduce the `CURE-TSR` dataset. Traffic sign images in the `CURE-TSR` dataset were cropped from the `CURE-TSD` dataset [95, 96, 97], which includes around 1.7 million real-world and simulator images. Real-world images were obtained from the BelgiumTS video sequences. There are 14 traffic signs with annotations in both environments, which are shown in Fig. Figure 3.4. Sign types include speed limit, goods vehicles, no overtaking, no stopping, no parking, stop, bicycle, hump, no left, no right, priority to, no entry, yield, and parking. Sequences were processed with state-of-the-art visual effect software Adobe(c) After Effects to simulate challenging conditions, which include rain, snow, haze, shadow, darkness, brightness, blurriness, dirtiness, colorlessness, sensor and codec errors. We visualize traffic sign images with 8 different challenge types and 5 different levels, which are used for our OOD detection experiments, in Figure 3.5. Level 5 images contain the most severe challenge effect and level 1 images are least affected by the challenging conditions. Since level 1 images are perceptually most similar to the challenge-free image, it is more challenging for anomaly detection algorithms to classify level 1 images as outliers.

We ensure the diversity challenging conditions and the proper scaling of challenge levels by performing baseline analysis with different features for traffic sign recognition. In the German traffic sign recognition benchmark (GTSRB) [93], histogram of oriented gradient (HOG) features were utilized to report the baseline traffic sign recognition results. In the Belgium traffic sign classification (BelgiumTSC) benchmark, cropped traffic sign images were converted into grayscale and rescaled to 28×28 patches, which were included

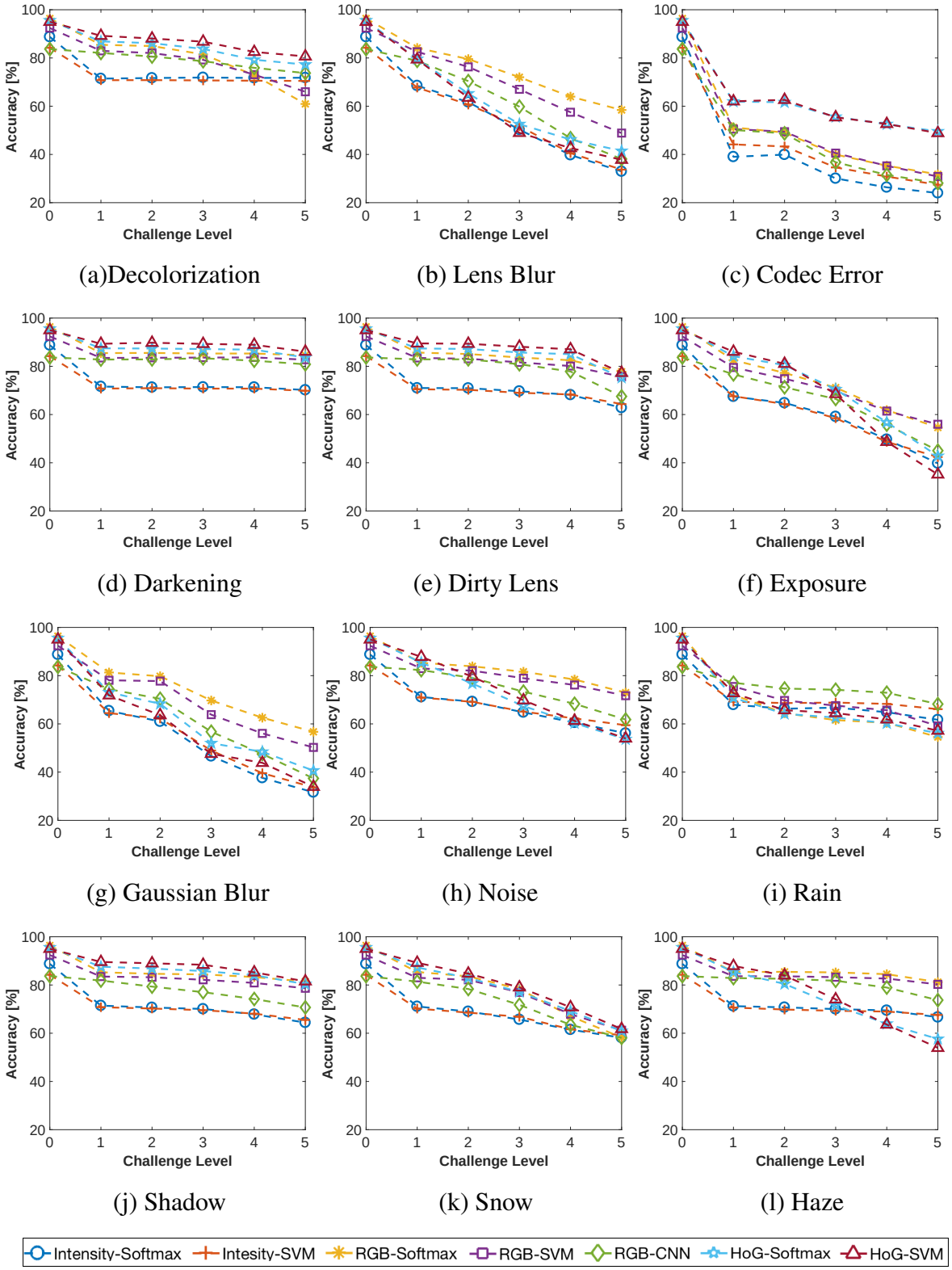


Figure 3.6: Performance versus challenge levels on CURE-TSR.

in the baseline. Moreover, HoG features were also used as a baseline method. They classified traffic sign images with methods including support vector machines (SVMs). Similar to GTSRB and BelgiumTS datasets, we use rescaled grayscale and color images as well as HoG features as baseline. In the final classification stage, we utilize one-vs-all SVMs with radial basis kernels and softmax classifiers. In addition to aforementioned techniques, we also use a shallow convolutional neural network, which consists of two convolutional layers followed by two fully connected layers, and a softmax classifier. We preprocessed images using l_2 normalization, mean subtraction, and division by standard deviation. We show the performance of these classifiers on 12 challenge types and 5 challenge conditions in Figure 3.6. As shown in the plots, the classifiers experience characteristic performance degradation across different challenging conditions and levels. This supports that CURE-TSR simulates diverse types of OOD and can be utilized as an effective benchmark for OOD detection tasks.

3.4 Experiments

3.4.1 Experimental Setup

We conduct controlled experiments to analyze the gradient features for OOD detection. In particular, we perform OOD class detection and OOD condition detection using gradients and compare the performance with activation-based representations. In OOD class detection, samples from one class are considered as inliers and other class samples are considered as outliers. For OOD condition detection, images without any effect are utilized as inliers and images captured under challenging conditions such as distortions or environmental effects are used as outliers. We only use the inliers for training and classify both inliers and outliers during testing.

For the fair comparison between gradients and activation-based representations, we first train a VAE as described in section 3.2 using the inliers. Then, we train three different classifiers with the same architecture using reconstruction error, latent loss, backpropagated

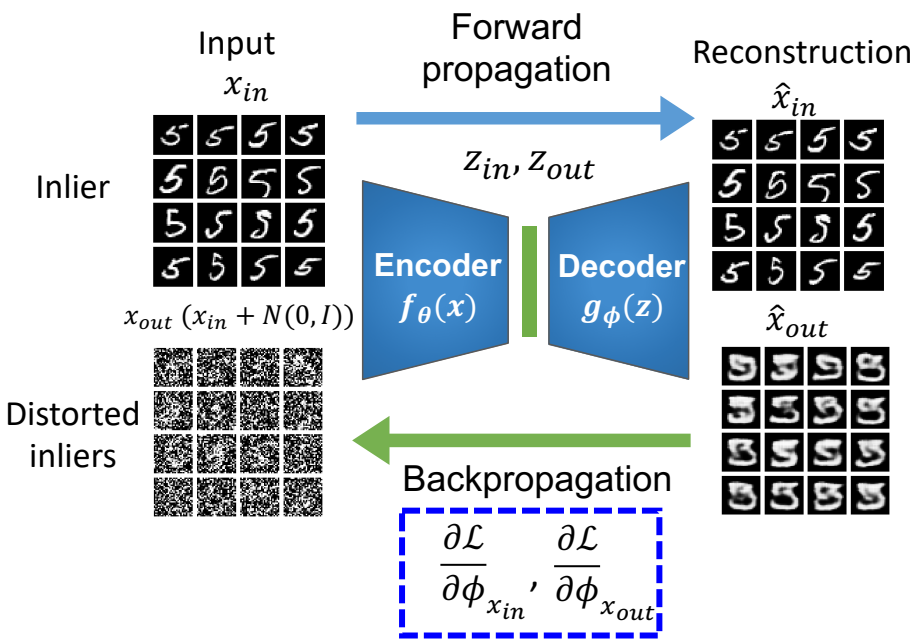


Figure 3.7: Generation of gradient features.

gradients as features. These classifiers are visualized in Figure 3.8. The classifier consists of two linear layers and sigmoid nonlinear activation layers between the linear layers. The reconstruction error and the latent loss are calculated as suggested in [17] but we do not take sum over all elements but use vectors as features for the classifiers. We obtain the gradients by backpropagating the reconstruction error as shown in Figure 3.7. To train a supervised classifier, we need outlier training images. As suggested in [45], we distort the inlier images by adding Gaussian noise and use the distorted images as the training outliers. For the OOD class detection, we extract gradients from the first layer of the decoder since the layer close to the latent representation is supposed to contain high-level semantic information. On the other hand, distortions or challenging conditions alter the low-level characteristics of images such as edges and colors. Therefore, we extract gradients from the last layer of the decoder for the OOD condition detection.

We use three image recognition datasets, which are MNIST [91], Fashion MNIST (fMNIST) [98], and CIFAR-10 [99], for the OOD class detection task and use CURE-TSR dataset [100] for the OOD condition detection. MNIST, fMNIST, and CIFAR-10 con-

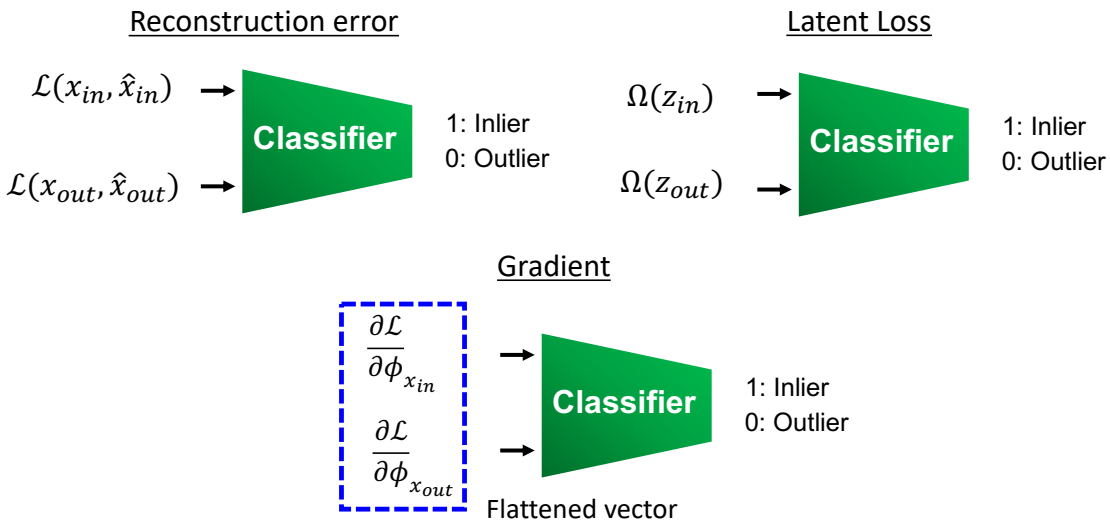


Figure 3.8: Three Classifiers trained with reconstruction error, latent loss, and gradient features.

Figure 3.9: Samples from benchmark image recognition datasets.

tain 10 classes of digits, fashion products and color objects, respectively. CURE-TSR contains traffic sign images with 12 challenging conditions and 5 challenge levels. We consider 5 challenging conditions which are `Lens blur`, `Dirty lens`, `Gaussian blur`, `Rain`, and `Haze` for this experiment. Test sets contain the same number of inliers and outliers. For MNIST and fMNSIT, we split the dataset into 5 folds and 60% of each class is used for training, 20% is used for validation, the remaining 20% is used for testing. For CIFAR-10 and CURE-TSR, the original training and test splits are used and 10% of the training images are held out for validation.

3.4.2 OOD class detection

In Table 3.1, we summarize the performance of the OOD class detectors trained using the activation-based representations (the reconstruction error and the latent loss) and the gra-

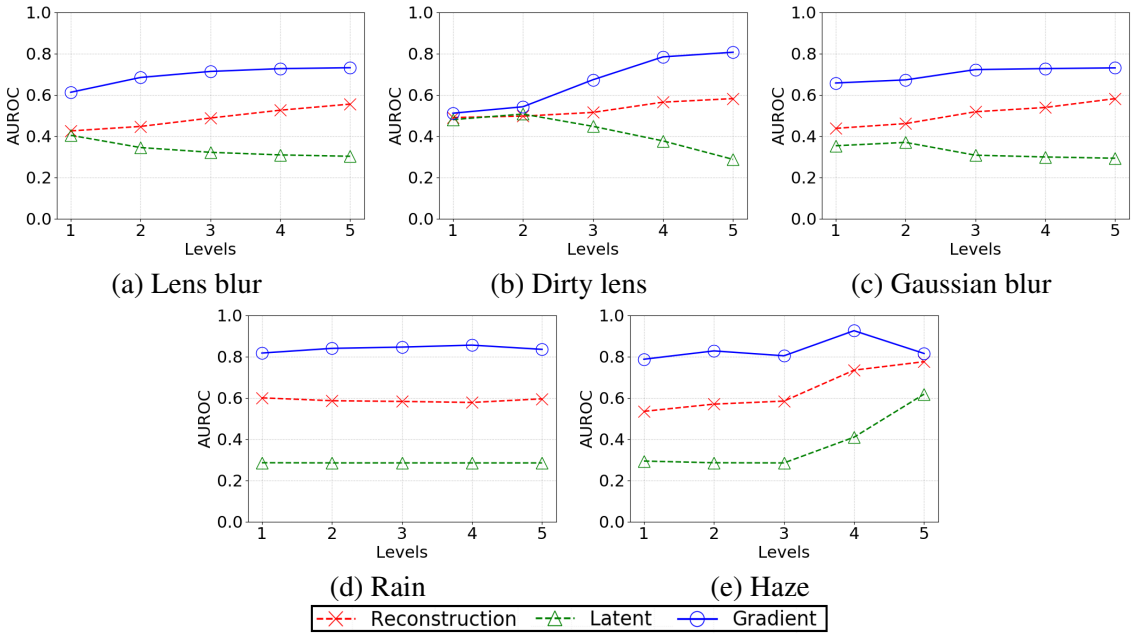


Figure 3.10: Novelty condition detection results on CURE-TSR.

dient. The performance is measured by area under receiver operation characteristic curve (AUROC) for each class and an average AUROC over different classes is also provided in the table. AUROC is bounded between 0 and 1 and the higher value indicates superior performance. As shown in the table, the classifiers trained on the gradients outperform those trained on the activation-based representations by a significant margin for almost all classes in three datasets. In particular, the best average AUROC performance obtained from the classifiers trained using the gradients is higher by 0.235, 0.1, and 0.02 respectively compared to the second best results in three datasets. Also, the variances of AUROC over 10 classes obtained from the gradients are 0.001, 0.003 in MNIST and fMNIST, respectively. These variances are significantly smaller than the second smallest variances 0.044 and 0.011, respectively for both datasets. In CIFAR-10, the variance of AUROC by the gradients is the second smallest. This indicates that different classes of anomalies are separated and characterized robustly using the backpropagated gradients. The reconstruction error shows particularly low performance on MNIST and this may be resulted from the fact that digit images take relatively small portion of the entire image and the setup of using the

Table 3.1: Novelty class detection results on MNIST, fMNIST, and CIFAR-10.

Dataset	Repre.	Classes										Average
		0	1	2	3	4	5	6	7	8	9	
MNIST	Recon.	0.043	0.916	0.293	0.132	0.103	0.158	0.101	0.115	0.291	0.147	0.230
	Latent	0.956	0.510	0.687	0.740	0.852	0.526	0.675	0.942	0.348	0.948	0.718
	Gradient	**0.985**	**0.994**	**0.941**	**0.928**	**0.953**	**0.926**	**0.980**	**0.960**	**0.894**	**0.968**	**0.953**
fMNIST	Recon.	0.778	0.952	0.831	0.799	0.801	0.787	0.748	0.939	0.610	0.932	0.818
	Latent	0.733	0.642	0.525	0.877	0.715	0.831	0.585	0.961	0.702	0.835	0.741
	Gradient	**0.913**	**0.958**	**0.883**	**0.922**	**0.907**	**0.924**	**0.798**	**0.974**	**0.925**	**0.975**	**0.918**
CIFAR-10	Recon.	0.600	0.485	0.539	**0.496**	0.532	0.444	0.601	**0.545**	0.634	0.541	0.542
	Latent	**0.683**	0.382	0.560	0.458	0.649	0.486	**0.724**	0.465	**0.662**	**0.550**	0.562
	Gradient	0.658	**0.543**	**0.632**	0.461	**0.725**	**0.493**	0.699	0.490	0.641	0.477	**0.582**

distorted inliers does not help learning tight decision boundary around the inliers. Given that the reconstruction error is backpropagated to generate the gradient features, we can understand the significance of directional information from the gradients by comparing the performance of the reconstruction errors and the gradients. In all three datasets, the performance from the gradient features outperforms that from the reconstruction error by at least 0.04 AUROC scores.

3.4.3 OOD condition detection

In Figure 3.10, we visualize the AUROC results over different challenge levels in each challenge type using CURE-TSR. The classifiers trained using the gradients outperform those trained on the reconstruction error and the latent loss for all challenge types and challenge levels. In terms of an average AUROC over challenge levels, the gradient shows the largest improvement in `Rain` followed by `Lens blur` and `Gaussian blur`. When the challenge level is low, challenge images are similar to challenge-free images and hard to detect. Except for `Dirty lens`, the gradients achieve at least 0.187 improvement over the second best results in all challenge types. Five challenging conditions are chosen to encompass acquisition imperfection, processing artifact, and environmental challenging conditions. The best results from the gradients show its representation capability in characterizing diverse types of challenging conditions.

3.5 Summary

In this chapter, we proposed a framework to characterize abnormality from the model perspective using gradients. We conducted a comprehensive analysis to compare the performance of OOD detection from the activation and the gradient. The statistical analysis demonstrates that the larger separation between inliers and outliers is achieved using the gradients compared to the activation. Also, we show that the classifiers trained using the gradients as features outperform those trained using common activation-based features in

OOD class and condition detection. Considering that most of existing works have only focused on developing descriptive activation-based representations, we leave using more sophisticated training schemes such as adversarial training with gradient features as remaining work for the future.

CHAPTER 4

CONSTRAINED GRADIENT-BASED REPRESENTATIONS

We develop a regularization technique for gradient-based representations to achieve accurate OOD detection. In the previous chapter, we show that raw gradients from the autoencoder are effective features to differentiate ID and OOD. Although the autoencoder constrains the activation-based representations through the reconstruction error, gradients which play a critical role in OOD detection remain unconstrained during training. We investigate the importance of modeling the normality for ID by constraining the gradients obtained from ID data. The normality is learned using an explicit gradient constraint that we developed. We motivate the gradient constraint from a theoretical interpretation of gradients. In addition, we show the constrained gradient-based representations achieve state-of-the-art performance for OOD class and condition detection in benchmark image recognition datasets. Also, we highlight the computational efficiency and the simplicity of the developed method in comparison with other state-of-the-art methods relying on adversarial networks or autoregressive models, which require at least 27 times more model parameters than the developed method.

4.1 Theoretical Interpretation of Gradients

We derive theoretical explanation for gradient-based representations from information geometry, particularly using the Fisher kernel. Based on the Fisher kernel, we show that the gradient-based representations characterize model updates from query data and differentiate normal from abnormal data. We utilize the same setup of an autoencoder described in section 3.1 but consider the encoder and the decoder as probability distributions [15]. Given the latent variable, z, the decoder models input distribution through a conditional distribution, $P_\phi(x|z)$. The autoencoder is trained to minimize the negative log-likelihood,

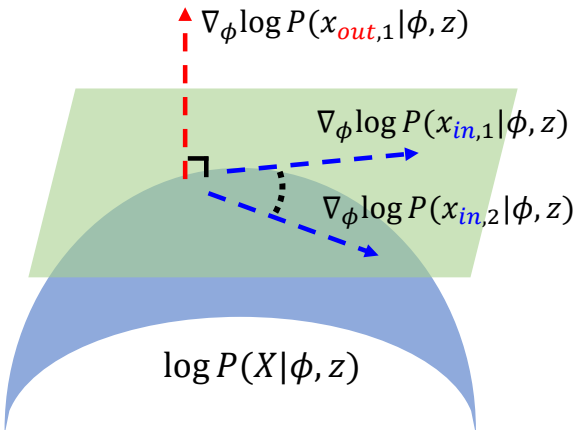


Figure 4.1: Gradient constraint on the manifold.

$-\log P_\phi(x|z)$. When x is a real value and $P_\phi(x|z)$ is assumed to be a Gaussian distribution, the decoder estimates the mean of the Gaussian. Also, the minimization of the negative log-likelihood corresponds to using a mean squared error as the reconstruction error. When x is a binary value, the decoder is assumed to be a Bernoulli distribution. The negative log-likelihood is formulated as a binary cross entropy loss. Considering the decoder as the conditional probability enables to interpret gradients using the Fisher kernel.

The Fisher kernel defines a metric between samples using the gradients of generative probability distribution [101]. Let X be a set of samples and $P(X|\theta)$ is a probability density function of the samples parameterized by $\theta = [\theta_1, \theta_2, ..., \theta_N]^\mathsf{T} \in \mathbb{R}^N$. This probability distribution models a Riemannian manifold with a local metric defined by Fisher information matrix, $F \in \mathbb{R}^{N \times N}$, as follows:

$$F = \mathop{\mathbb{E}}_{x \in X}[U_\theta^X {U_\theta^X}^\mathsf{T}] \quad \text{where} \quad U_\theta^X = \nabla_\theta \log P(X|\theta). \tag{4.1}$$

U_θ^X is called the Fisher score which describes the contribution of the parameters in modeling the data distribution. In [101], the authors propose the Fisher kernel to measure the difference between two samples based on the Fisher score. The Fisher kernel, K_{FK}, is defined as

$$K_{FK}(X_i, X_j) = {U_\theta^{X_i}}^\mathsf{T} F^{-1} U_\theta^{X_j}, \tag{4.2}$$

where X_i and X_j are two data samples. The Fisher kernels enable to extract discriminant features from the generative model and they have been actively used in diverse applications such as image categorization, image classification, and action recognition [59], [102], [103].

We use the Fisher kernel estimated from the autoencoder for abnormal OOD detection. The distribution of the decoder is parameterized by the weights, ϕ, and the Fisher score from the decoder, $U^X_{\phi,z}$, is defined as

$$U^X_{\phi,z} = \nabla_\phi \log P(X|\phi, z). \tag{4.3}$$

Also, since the distribution is learned to be generalizable to the test data, we can use the Fisher kernel to measure the distance between training data and normal test data, and between training data and abnormal test data. The Fisher kernel for normal data (inliers), K^{in}_{FK}, and abnormal data (outliers), K^{out}_{FK}, are derived as follows, respectively:

$$K^{in}_{FK}(X_{tr}, X_{te,in}) = {U_\phi}^{X_{tr}\mathsf{T}} F^{-1} U^{X_{te,in}}_{\phi,z} \tag{4.4}$$

$$K^{out}_{FK}(X_{tr}, X_{te,out}) = {U_\phi}^{X_{tr}\mathsf{T}} F^{-1} U^{X_{te,out}}_{\phi,z}, \tag{4.5}$$

where $X_{tr}, X_{te,in}, X_{te,out}$ are training data, normal test data, and abnormal test data, respectively. For ideal anomaly detection, K^{out}_{FK} should be larger than K^{in}_{FK} to clearly differentiate normal and abnormal data. The difference between K^{in}_{FK} and K^{out}_{FK} is characterized by the Fisher scores $U^{X_{te,in}}_{\phi,z}$ and $U^{X_{te,out}}_{\phi,z}$. Therefore, the Fisher scores from query data are discriminant features for detecting anomalies. We propose to estimate the Fisher scores using the backpropagated gradients with respect to the weights of the decoder. Since the autoencoder is trained to minimize the negative log-likelihood loss, $\mathcal{L} = -\log P_\phi(x|z)$, the backpropagated gradients, $\frac{\partial \mathcal{L}}{\partial \phi}$, obtained from normal and abnormal data estimate $U^{X_{te,in}}_{\phi,z}$ and $U^{X_{te,out}}_{\phi,z}$

when the autoencoder is trained with a sufficiently large amount of data to model the data distribution. Therefore, we can interpret the gradient-based representations as discriminant representations obtained from the conditional probabilistic modeling of data for anomaly detection.

We visualize the gradients with respect to the weights of the decoder obtained by back-propagating the reconstruction error, $\mathcal{L}$, from normal data, $x_{in,1}, x_{in,2}$, and abnormal data, $x_{out,1}$, in Figure 4.1. These gradients estimate the Fisher scores for inliers and outliers, which need to be clearly separated for anomaly detection. Given the definition of the Fisher scores, the gradients from normal data should contribute less to the change of the manifold compared to those from abnormal data. Therefore, the gradients from normal data should reside in the tangent space of the manifold but abnormal data results in the gradients orthogonal to the tangent space. We achieve this separation in gradient-based representations through directional constraint described in the following section.

4.2 Gradient Constraint

The separation between inliers and outliers in the representation space is often achieved by modeling the normality of data. The deviation from the normality model captures the abnormality. The normality is often modeled through constraints imposed during training. The constraint allows normal data to be easily constrained but makes abnormal data deviates. For example, the autoencoders constrain the output to be similar to the input and the reconstruction error measures the deviation. A variational autoencoder (VAE) [104] and an adversarial autoencoder (AAE) [48] often constrain the latent representation to follow the Gaussian distribution and the deviation from the Gaussian distribution characterizes anomalies. In the gradient-based representations, we also impose a constraint during training to model the normality of data and further differentiate $U_{\phi,z}^{X_{te,in}}$ from $U_{\phi,z}^{X_{te,out}}$ defined in section 4.1.

We propose to train an autoencoder with a directional gradient constraint to model the

normality. In particular, based on the interpretation of gradients from the Fisher kernel perspective, we enforce the alignment between gradients. This constraint makes the gradients from normal data aligned with each other and result in small changes to the manifold. On the other hand, the gradients from abnormal data will not be aligned with others and guide abrupt changes to the manifold. We utilize a gradient loss, $\mathcal{L}_{grad}$, as a regularization term in the entire loss function, J. We calculate the cosine similarity between the gradients of a certain layer i in the decoder at the k^{th} iteration of training, $\frac{\partial \mathcal{L}}{\partial \phi_i}^k$, and the average of the training gradients of the same layer i obtained until the $(k-1)^{th}$ iteration, $\frac{\partial \mathcal{J}}{\partial \phi_i}_{avg}^{k-1}$. The gradient loss at the k^{th} iteration of training is obtained by averaging the cosine similarity over all the layers in the decoder as follows:

$$\mathcal{L}_{grad} = -\mathop{\mathbb{E}}_{i}\left[\text{cosSIM}\left(\frac{\partial \mathcal{J}}{\partial \phi_i}_{avg}^{k-1}, \frac{\partial \mathcal{L}}{\partial \phi_i}^k\right)\right], \quad \text{where} \quad \frac{\partial \mathcal{J}}{\partial \phi_i}_{avg}^{k-1} = \frac{1}{(k-1)}\sum_{t=1}^{k-1}\frac{\partial \mathcal{J}}{\partial \phi_i}^t. \quad (4.6)$$

The overall loss, J, is defined as

$$J = \mathcal{L} + \Omega + \alpha\mathcal{L}_{grad}. \quad (4.7)$$

The first and the second terms are the reconstruction error and the latent loss, respectively and they are defined by different types of autoencoders. α is a weight for the gradient loss. We set sufficiently small α value to ensure that gradients actively explore the optimal weights until the reconstruction error and the latent loss become small enough. Based on the interpretation of the gradients described in section 4.1, we only constrain the gradients of the decoder layers and the encoder layers remain unconstrained.

During training, $\mathcal{L}$ is first calculated from the forward propagation. Through the backpropagation, $\frac{\partial \mathcal{L}}{\partial \phi_i}^k$ is obtained without updating the weights. Based on the obtained gradient, the entire loss J is calculated and finally the weights are updated using backpropagated gradients from the loss J. An anomaly score is defined by the combination of the reconstruction error and the gradient loss as $\mathcal{L} + \beta\mathcal{L}_{grad}$. Although we use α to weight the gradient

loss during training, we found that the gradient loss is often more effective than the reconstruction error for anomaly detection. To better balance the two losses, we use $\beta = 4\alpha$ for all the experiments and show that the weighted combination of two losses improve the performance. The proposed anomaly detection algorithm using **Grad**ient **Con**straint is called GradCon.

4.3 Experiments

4.3.1 Experimental Setup

We conduct abnormal OOD detection experiments to both qualitatively and quantitatively evaluate the performance of the gradient-based representations. In particular, we perform OOD class detection and OOODO condition detection using the gradient constraint and compare `GradCon` with other state-of-the-art activation-based OOD detection algorithms. In OOD class detection, images from one class of a dataset are considered as inliers and used for the training. Images from other classes are considered as outliers. In OOD condition detection, images without any effect are utilized as inliers and images captured under challenging conditions such as distortions or environmental effects are considered as outliers. Both inliers and outliers are given to the network during testing. The OOD detection algorithms are expected to correctly classify data of which class and condition differ from those of the training data.

Datasets We utilize four benchmark datasets, which are CIFAR-10 [99], MNIST [91], fashion MNIST (fMNIST) [98], and CURE-TSR [100] to evaluate the performance of the proposed algorithm. We use CIFAR-10, MNIST, fMNIST for OOD class detection and CURE-TSR for OOD condition detection. CIFAR-10 dataset consists of 60,000 color images with 10 classes. MNIST dataset contains 70,000 handwritten digit images from 0 to 9 and fMNIST dataset also has 10 classes of fashion products and there are 7,000 images per class. CURE-TSR dataset has $637,560$ color traffic sign images which consist of 14 traffic sign types under 5 levels of 12 different challenging conditions. For CIFAR-10,

CURE-TSR, and MNIST, we follow the protocol described in [105] to create splits. To be specific, we utilize the original training and the test split of each dataset for training and testing. 10% of training images are held out for validation. For fMNIST, we follow the protocol described in [47]. The dataset is split into 5 folds and 60% of each class is used for training, 20% is used for validation, the remaining 20% is used for testing. In the experiments with CIFAR-10, MNIST, and fMNIST, we use images from one class as inliers for training. During testing, inlier images and the same number of oulier images randomly sampled from other classes are utilized. For CURE-TSR, challenge-free images are utilized as inliers for training. During testing, challenge-free images are utilized as inliers and the same images with challenging conditions are utilized as outliers. We particularly use 5 challenge levels with 8 challenging conditions which are `Decolorization`, `Lens blur`, `Dirty lens`, `Exposure`, `Gaussian blur`, `Rain`, `Snow`, and `Haze`. All the results are obtained using area under receiver operation characteristic curve (AUROC) and we also report F1 score in fMNIST dataset for the fair comparison with the state-of-the-art method [47].

Implementation details We train a convolutional autoencoder (CAE) for `GradCon`. The encoder and the decoder consist of 4 convolutional layers and the dimension of the latent variable is $3 \times 3 \times 64$. The number of convolutional filters for each layer in the encoder is 32, 32, 64, 64 and the kernel size is 4×4 for all the layers. The architecture of the decoder is symmetric to the encoder. Adam optimizer [106] with the learning rate of 0.001 is used for training. We use mean square error as the reconstruction error and do not use any latent loss for the CAE ($\Omega = 0$). $\alpha = 0.03$ is used to weight the gradient loss.

4.3.2 Baseline Comparison

We compare the performance of the gradient-based representations in characterizing OOD data with the activation-based representations. Furthermore, we show that the gradient-based representations can complement the activation-based representations and improve

the performance of OOD detection. We train four different autoencoders, which are CAE, CAE with the gradient constraint (CAE + Grad), VAE, VAE with the gradient constraint (VAE + Grad) for the baseline experiments. VAEs are trained using binary cross entropy as the reconstruction error and Kullback Leibler (KL) divergence as the latent loss. Implementation details for VAEs are same as those for the CAE described in subsection 4.3.1. We train the autoencoders using images from each class of CIFAR-10. Two losses defined by the activation-based representations, which are the reconstruction error (Recon) and the latent loss (Latent), and the gradient loss (Grad) defined by the gradient-based representations are separately used as anomaly scores for detection. AUROC results are reported in Table 4.1 and the highest AUROC for each class is highlighted in bold.

Effectiveness of the gradient constraint (CAE vs. CAE+Grad) We first compare the performance of CAE and CAE + Grad to analyze the effectiveness of the gradient-based representation with constraint. The reconstruction error from CAE and CAE + Grad achieves comparable average AUROC scores. The gradient loss from CAE + Grad achieves the best performance with an average AUROC of 0.661. This shows that the gradient constraint marginally sacrifices the performance from the activation-based representation and achieve the superior performance from the gradient-based representation.

Performance sacrifice from the latent constraint (CAE vs. VAE) We evaluate the effect of the latent constraint by comparing CAE and VAE. The latent loss of VAE achieves the improved performance compared to the reconstruction error of CAE by an average AUROC of 0.019. However, the performance of the reconstruction error from VAE is lower than that from CAE by 0.038. This shows that the latent constraint sacrifices the performance from another activation-based representation which is the reconstructed image. Since both latent representation and reconstructed image are obtained from forward propagation, the constraint imposed in the latent space affects the reconstruction performance. Therefore, using a combination of multiple activation-based representations faces limitations in improving the performance.

Table 4.1: Baseline anomaly detection results on CIFAR-10. The reconstruction error (Recon) and the latent loss (Latent) are obtained from the activation-based representations and the gradient loss (Grad) is obtained from the gradient-based representations.

Model	Loss	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	Average
CAE	Recon	0.682	0.353	0.638	0.587	0.669	**0.613**	0.495	0.498	0.711	0.390	0.564
CAE	Recon	0.659	0.356	**0.640**	0.555	0.695	0.554	0.549	0.478	0.695	0.357	0.554
+ Grad	Grad	**0.752**	0.619	0.622	0.580	0.705	0.591	0.683	**0.576**	**0.774**	**0.709**	**0.661**
VAE	Recon	0.553	0.608	0.437	0.546	0.393	0.531	0.489	0.515	0.552	0.631	0.526
	Latent	0.634	0.442	**0.640**	0.497	**0.743**	0.515	**0.745**	0.527	0.674	0.416	0.583
VAE	Recon	0.556	0.606	0.438	0.548	0.392	0.543	0.496	0.518	0.552	0.631	0.528
+ Grad	Latent	0.586	0.396	0.618	0.476	0.719	0.474	0.698	0.537	0.586	0.413	0.550
	Grad	0.736	**0.625**	0.591	**0.596**	0.707	0.570	0.740	0.543	0.738	0.629	0.647

Table 4.2: Anomaly detection results from the gradients of each layer in the decoder.

Layer		1^{st}	2^{nd}	3^{rd}	4^{th}	All
CIFAR-10		**0.648**	**0.649**	0.628	0.605	**0.661**
CURE -TSR	DL	**0.688**	0.640	0.649	**0.681**	**0.708**
	EX	**0.859**	0.811	0.781	**0.833**	**0.891**
	SN	**0.677**	0.612	0.628	**0.693**	**0.702**

Complementary features from the gradient constraint (VAE vs. VAE +Grad) Comparison between VAE and VAE + Grad shows the effectiveness of using the gradient constraint with the activation constraint. The gradient loss in VAE + Grad achieves the second best average AUROC and outperforms the latent loss in the VAE by 0.064. The performance from the reconstruction error is comparable between VAE and VAE + Grad. The average AUROC of the latent loss from VAE + Grad is marginally sacrificed by 0.033 compared to that from VAE. In both CAE + Grad and VAE + Grad, the performance gain from the gradient loss is always greater than the sacrifice in other activation-based representations. This is contrary to the CAE and VAE comparison where the performance gain is smaller than the sacrifice from the reconstruction error. Since gradients are obtained in parallel with the activation, constraining gradients less affects the OOD detection performance from the activation-based representations. Thus, the gradient-based representations can provide complementary features to the activation-based representations for OOD detection.

OOD condition detection We further analyze the discriminant capability of the gradient-based representations for diverse challenging conditions and levels. We compare the performance of CAE and CAE + Grad using the reconstruction error (Recon) and the gradient loss (Grad). Samples with challenging conditions and the AUROC performance are visualized in Figure 4.2. For all challenging conditions and levels, CAE + Grad achieves the best performance. In particular, except for snow level $1\sim3$, the gradient loss achieves the best performance and for snow level $1\sim3$, the reconstruction error of CAE + Grad achieves the best performance. In terms of the average AUROC over challenge levels, the gradient loss of CAE + Grad outperforms the reconstruction error of CAE by the largest margin of 0.612 in rain and the smallest margin of 0.089 in snow. These test conditions encompass acquisition imperfection, processing artifact, and environmental challenging conditions. The superior performance of the gradient loss shows that the gradient-based representation effectively characterizes diverse types and levels of unseen challenging conditions.

Decomposition of the gradient loss We decompose the gradient loss and analyze the con-

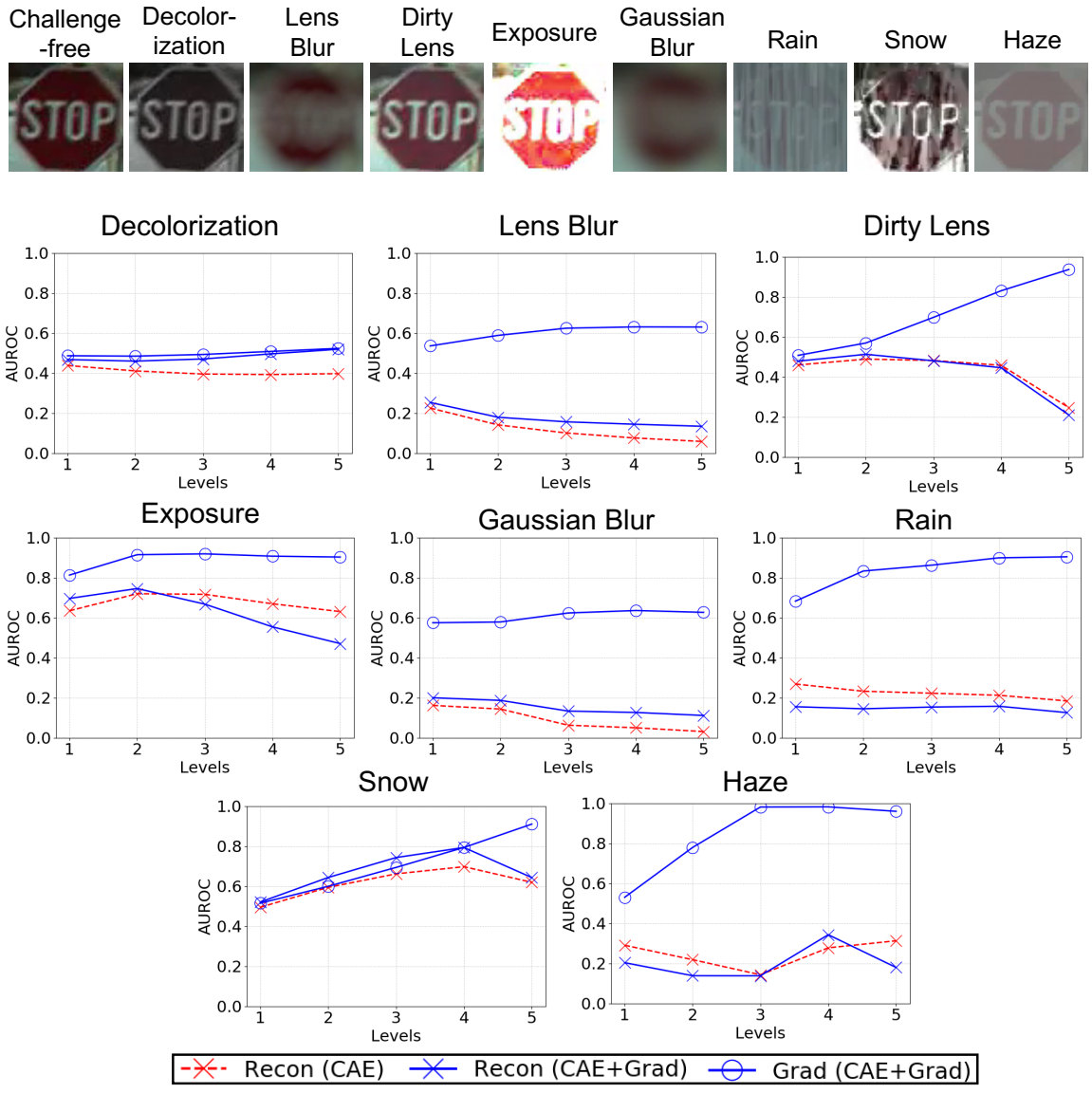


Figure 4.2: Baseline anomaly detection results on CURE-TSR.

Table 4.3: Anomaly detection AUROC results on CIFAR-10.

	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	Average
OCSVM [38]	0.630	0.440	0.649	0.487	0.735	0.500	0.725	0.533	0.649	0.508	0.586
KDE [107]	0.658	0.520	**0.657**	0.497	0.727	0.496	**0.758**	0.564	0.680	0.540	0.610
DAE [108]	0.411	0.478	0.616	0.562	0.728	0.513	0.688	0.497	0.487	0.378	0.536
VAE [104]	0.634	0.442	0.640	0.497	**0.743**	0.515	0.745	0.527	0.674	0.416	0.583
PixelCNN [109]	**0.788**	0.428	0.617	0.574	0.511	0.571	0.422	0.454	0.715	0.426	0.551
LSA [44]	0.735	0.580	**0.690**	0.542	**0.761**	0.546	**0.751**	0.535	0.717	0.548	0.641
AnoGAN [110]	0.671	0.547	0.529	0.545	0.651	0.603	0.585	**0.625**	0.758	0.665	0.618
DSVDD [111]	0.617	**0.659**	0.508	**0.591**	0.609	**0.657**	0.677	**0.673**	0.759	**0.731**	0.648
OCGAN [105]	0.757	0.531	0.640	**0.620**	0.723	**0.620**	0.723	0.575	**0.820**	0.554	**0.657**
GradCon	**0.760**	**0.598**	0.648	0.586	0.733	0.603	0.684	0.567	**0.784**	**0.678**	**0.664**

Table 4.4: Anomaly detection AUROC results on MNIST.

	0	1	2	3	4	5	6	7	8	9	Average
OCSVM [38]	0.988	**0.999**	0.902	0.950	0.955	0.968	0.978	0.965	0.853	0.955	0.951
KDE [107]	0.885	0.996	0.710	0.693	0.844	0.776	0.861	0.884	0.669	0.825	0.814
DAE [108]	0.894	**0.999**	0.792	0.851	0.888	0.819	0.944	0.922	0.740	0.917	0.877
VAE [104]	**0.997**	**0.999**	0.936	0.959	**0.973**	0.964	**0.993**	0.976	0.923	**0.976**	0.970
PixelCNN [109]	0.531	0.995	0.476	0.517	0.739	0.542	0.592	0.789	0.340	0.662	0.618
LSA [44]	0.993	**0.999**	**0.959**	**0.966**	0.956	0.964	**0.994**	**0.980**	**0.953**	**0.981**	**0.975**
AnoGAN [110]	0.966	0.992	0.850	0.887	0.894	0.883	0.947	0.935	0.849	0.924	0.913
DSVDD [111]	0.980	**0.997**	0.917	0.919	0.949	0.885	0.983	0.946	**0.939**	0.965	0.948
OCGAN [105]	**0.998**	**0.999**	0.942	0.963	**0.975**	**0.980**	0.991	**0.981**	**0.939**	**0.981**	**0.975**
GradCon	0.995	**0.999**	**0.952**	**0.973**	0.969	**0.977**	**0.994**	0.979	0.919	0.973	**0.973**

tribution of gradients from each layer on anomaly detection. Instead of the gradient loss obtained by averaging the cosine similarity over all the layers as Equation 4.6, we use the cosine similarity from each layer as an anomaly score. The average AUROC results obtained by the gradients from the first to the fourth layer of the decoder are reported in Table 4.2. Also, results obtained by averaging the cosine similarity over all layers are reported. We use CIFAR-10 and `Dirty Lens` (DL), `Exposure` (EX), `Snow` (SN) challenge types of CURE-TSR. In CIFAR-10, inlier class and outlier classes share most of low-level features such as edges or colors. Also, semantic information mostly differentiate classes. Since the layers close to the latent space focus more on high-level characteristics of data, the gradient loss from the first and the second layer show the largest contribution on OOOD detection. In CURE-TSR, challenging conditions alter low-level characteristics of images such as edges or colors. Therefore, the last layer of the decoder also contributes more than middle layers for OOD condition detection. This shows that gradients extracted from different layers characterize abnormality at different levels of data abstraction. In both datasets, results obtained by combining all the layers (All) show the best performance. Given that losses defined by activation-based representations can be calculated only from the output of specific layers, using gradients from all the layers enable to capture abnormality in both low-level and high-level characteristics of data.

4.3.3 Comparison With State-of-The-Art Algorithms

We evaluate the performance of GradCon which uses the combination of the reconstruction error and the gradient loss as an anomaly score. We compare GradCon with other benchmarking and state-of-the-art algorithms. The AUROC results on CIFAR-10 and MNIST are reported in Table 4.3 and Table 4.4, respectively. Top two AUROC scores for each class are highlighted in bold. GradCon achieves the best average AUROC performance in CIFAR-10 while achieving the second best performance in MNIST by the gap of 0.002. In Figure 4.3, we visualize the histogram of the reconstruction error, the latent loss, and the

gradient loss for inliers and outliers to further analyze the state-of-the-art performance of the proposed method. We calculate each loss for all the inliers and the outliers in MNIST. Also, we provide the percentage of overlap calculated by dividing the number of samples in the overlapped region of the histograms by the total number of samples. Ideally, measured errors on each representation should separate the histograms of inliers and outliers as much as possible for effective OOD detection. The gradient loss achieves the least number of samples overlapped which explains the state-of-the art performance achieved by GradCon. For CIFAR-10, we perform the same histogram analysis and histograms are visualized in Figure 4.4. The gradient loss shows the smallest overlap compared to other two losses defined in activation-based representations. This statistical analysis also supports the superior performance of GradCon compared to other reconstruction error or latent loss-based algorithms reported in Table 4.3.

Comparison between histograms from MNIST visualized in Figure 4.3 and those from CIFAR-10 shows that the gradient loss is more effective when data becomes complicated and challenging for anomaly detection. In MNIST, simple low-level features such as curved edges or straight edges can be class discriminant features for OOD detection. On the other hand, CIFAR-10 contains images with richer structure and features than MNIST. Therefore, ID and OOD data are not easily separable and the overlap between histograms is significantly larger in CIFAR-10 than MNIST. In CIFAR-10, the overlap of the gradient loss is smaller than the second smallest overlap of the reconstruction error by 12.4%. In MNIST, the overlap of the gradient loss is smaller than the second smallest overlap by 5.7%. GradCon also outperforms other state-of-the-art methods by a larger margin of AUROC in CIFAR-10 compared to MNIST. The overlap and performance differences show that the contribution of the gradient loss becomes more significant when data is complicated and challenging for anomaly detection.

we report the average AUROC performance of GradCon in comparison with that of additional benchmarking and state-of-the-art algorithms using fMNIST in Table 4.5. Grad-

Table 4.5: Average AUROC result of GradCon compared with benchmarking and state-of-the-art anomaly detection algorithms on fMNIST.

Method	ALOCCDR [45]	ALOCCD [45]	DCAE [41]
AUROC	0.753	0.601	0.908
Method	OCGAN [105]	GPND [47]	GradCon
AUROC	0.924	0.933	0.934

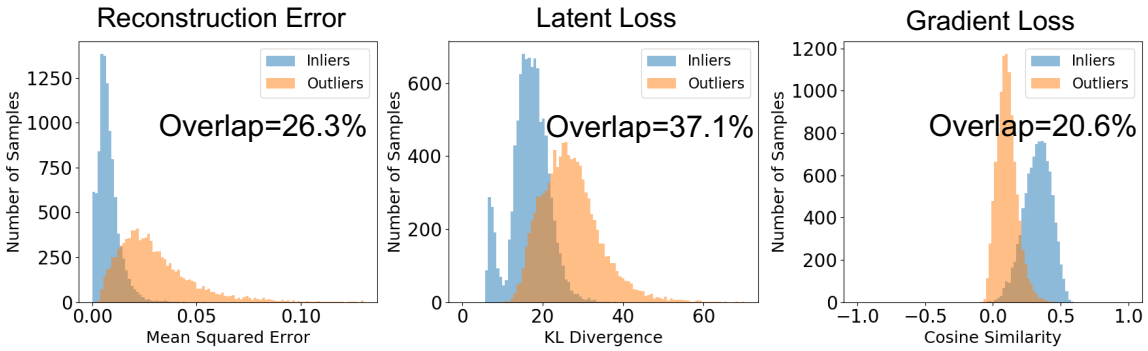


Figure 4.3: Histogram analysis on activation losses and gradient loss in MNIST.

Con outperforms all the compared algorithms including GPND. Given that ALOCC, OCGAN, and GPND are all based on adversarial training to further constrain the activation-based representations, GradCon achieves the best performance in fMNIST only based on a CAE and requires significantly less computations.

4.3.4 Ablation Study

We perform ablation study to analyze the performance of GradCon in comparison with the second best state-of-the-art algorithm denoted as GPND [47] in fMNIST. In this fMNIST experiment, we change the ratio of outliers in the test set from 10% to 50% and evaluate the performance in terms of AUROC and F1 score. We report the results from the gradient loss (Grad) and GradCon in Table Table 4.6. GradCon outperforms GPND in all outlier ratios in terms of AUROC. Except for the 10% of outlier ratio, GradCon achieves higher F1 scores than GPND. The results of the gradient loss and GradCon show that the combination of the gradient loss and the reconstruction error improves the performance for all the outlier ratios in terms of AUROC and F1 score.

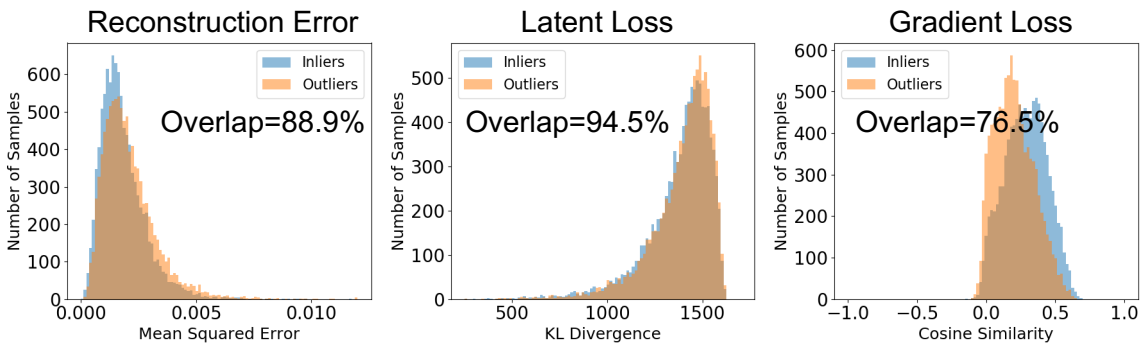


Figure 4.4: Histogram analysis on activation losses and gradient loss in CIFAR-10.

Table 4.6: Anomaly detection results on fMNIST.

% of outlier		10	20	30	40	50
F1	GPND	**0.968**	**0.945**	0.917	0.891	0.864
	Grad	0.964	0.939	0.917	0.899	0.870
	GradCon	0.967	**0.945**	**0.924**	**0.905**	**0.871**
AUC	GPND	0.928	0.932	0.933	0.933	0.933
	Grad	0.931	0.925	0.926	0.928	0.926
	GradCon	**0.938**	**0.933**	**0.935**	**0.936**	**0.934**

GradCon requires significantly less computational resources compared to other state-of-the-art algorithms. To show the computational efficiency of GradCon, we measure the average inference time per image using a machine with two GTX Titan X GPUs and compare computation time. While the average inference time per image for GPND on fMNIST is 5.72 *ms*, GradCon takes only 3.08 *ms* which is around 1.9 time faster. Also, we compare the number of model parameters for GradCon with that for the state-of-the-art algorithms in Table 4.7. AnoGAN, GPND, and LSA are based on a GAN [18], an AAD [48], and an autoregressive model [112], respectively but GradCon is solely based on a CAE. Hence, the number of model parameters for GradCon is approximately $27, 29, 59$ times less than that for AnoGAN, GPND, and LSA, respectively. Most of the state-of-the-art algorithms require additional training of adversarial networks or probabilistic modeling on top of the activation-based representations from the encoder and the decoder. Since GradCon is only based on the reconstruction error and the gradient loss of the CAE, it is computationally efficient even while achieving the state-of-the-art performance.

Table 4.7: Number of model parameters required to be trained for GradCon and other state-of-the-art methods.

Method	AnoGAN	GPND	LSA	GradCon
# of parameters	6,338,176	6,766,243	13,690,160	230,721

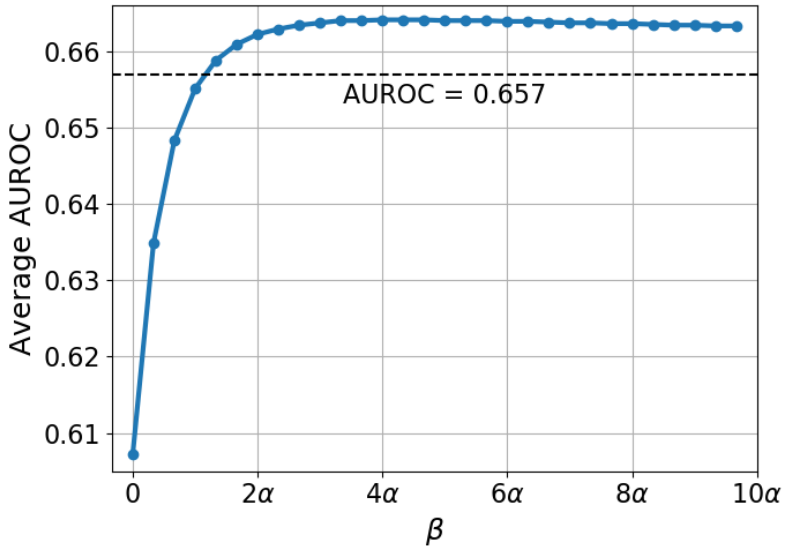


Figure 4.5: Average AUROC results with different β parameters in CIFAR-10. $\alpha = 0.03$ is utilized to train the CAE. The dotted line (average AUROC = 0.657) indicates the performance of OCGAN which achieves the second best performance in CIFAR-10.

Finally, we analyze the impact of different parameter settings on the performance of GradCon. The final anomaly score of GradCon is given as $\mathcal{L} + \beta\mathcal{L}_{grad}$, where $\mathcal{L}$ is the reconstruction error and $\mathcal{L}_{grad}$ is the gradient loss. While we use α parameter to weight the gradient loss and constrain the gradients during training, we observe that the gradient loss generally shows better performance as an anomaly score than the reconstruction error. Hence, we use $\beta = n\alpha$, where n is constant, to weight the gradient loss more for the anomaly score. We evaluate the average AUROC performance of GradCon with different β parameters using CIFAR-10 in Figure 4.5. In particular, we change the scaling constant, n, to change β in the x-axis of the plot. The performance of GradCon improves as we increase β in the range of $\beta = [0, 2\alpha]$. Also, GradCon consistently achieves state-of-the-art performance across a wide range of β parameter settings when $\beta \geq 1.67\alpha$. To be specific, GradCon always outperforms OCGAN which achieves the second best average

AUROC performance of 0.657 in CIFAR-10 when $\beta \geq 1.67\alpha$. This analysis shows that GradCon achieves the best performance in CIFAR-10 across a wide range of β.

4.4 Summary

We propose using a gradient-based representation for OOD detection by characterizing model behavior on anomalies. We introduce the theoretical interpretation of gradients and derive an anomaly score based on the deviation of gradients from the directional constraint. From thorough baseline analysis, we show the effectiveness of gradient-based representations for anomaly detection in comparison with the activation-based representations. Also, the proposed anomaly detection algorithm, GradCon, which is the combination of the reconstruction error and the gradient loss achieves the state-of-the-art performance in benchmarking image recognition datasets. In terms of the computational efficiency, GradCon has significantly less number of model parameters and shows faster inference time compared to other state-of-the-art anomaly detection algorithms. Given that most of anomaly detection algorithms adopt adversarial training frameworks or probabilistic modelings on activation-based representations, using more sophisticated training frameworks on gradient-based representations remains for future work.

CHAPTER 5
ACTIVATION-BASED REPRESENTATIONS LEARNED WITH AUXILIARY INFORMATION

In this chapter, we develop activation-based representations learned with auxiliary information to complement the limitations of activation-based and gradient-based representations. We first explain the advantages and the limitations of activation-based and gradient-based representations. As shown in the previous chapters, gradient-based representations have a clear advantage in characterizing OOD over the activation-based representations. However, the gradient-based representations also face challenges when they are used for generalizing to OOD with complicated ID data. To overcome this limitation, we learn aligned activation-based representations for both visual data and auxiliary information for detecting and generalizing to OOD. The auxiliary information such as class attributes can be easily obtained for both ID and OOD data. Therefore, auxiliary information of ID and OOD are projected in the same representation of visual data and used to differentiate ID from OOD. We validate OOD detection capability of activation-based representations learned with auxiliary information using four image recognition datasets with diverse granularity. We show that the activation-based representations learned with auxiliary information can complement the limitations of gradient-base representations and achieve better OOD detection performance.

5.1 Limitations of Gradient-based Representations

Gradient-based representations face challenges when they are used for both detecting and generalizing to OOD. An ideal representation should be capable of both detecting and generalizing to OOD. To do so, the representation should not only differentiate ID from OOD but also characterize data for the target task to be easily performed. For instance, when the

target task is image recognition, we first expect ID and OOD data are clearly distinguishable in the representation space. Also, representations should be clustered by minimizing intra-class distance while maximizing inter-class distance for ID and OOD, respectively. The gradient-based representations effectively differentiate ID from OOD. However, gradients only capture orthogonal components of the learned manifold where useful clusters for the image recognition are formed. Since the target task often requires knowledge learned from training data, which are characterized by the activation-based representations, the gradient-based representations are limited to perform both generalization tasks and OOD detection.

In addition, the gradient constraint limits the capability of activation-based representations for the generalization tasks. The gradient constraint is an essential component for the gradient-based representations to distinguish between ID and OOD. However, it explicitly enforces the direction of gradients which limits the neural network to explore more effective activation-based representations for the target tasks. This limitation can be better understood from Figure 5.1 (a). We visualize the gradient from ID data, $\nabla_\phi \log P(x_{in}|\phi, z)$, and OOD data, $\nabla_\phi \log P(x_{out}|\phi, z)$, on a two-dimensional plane manifold. We follow the notations define in section 4.1. The gradient constraint enforces the gradients from the reconstruction error to be aligned with the average of all past gradients. This constrains the direction of gradients to be tangential to the manifold and prevents the manifold from expanding to the orthogonal direction. Given that the activation-based representations reside in the learned manifold, the constraint in the learned manifold limits the representation capability of activation-based representations for the target generalization tasks.

Finally, constraining gradients becomes challenging when ID data is diverse and complicated. Small size images with limited visual features do not require a large amount of learning capacity to represent them. This means that the model updates are limited into a small subspace and gradients can be easily constrained. However, when ID data becomes diverse, the model requires updates that result in a complex learned manifold. In this case, gradients are more diverging and it become challenging to constrain them. In addition,

most of state-of-the-art algorithms for vision applications rely on features extracted from ImageNet-pretrained deep neural networks such as a ResNet-101 [12]. When the model utilizes the pre-extracted features, the model does not have a chance to learn constraining the gradients directly from the image, which causes the gradient constraint to be ineffective.

We propose complementing aforementioned limitations by using activation-based representations with auxiliary information. From the gradient-based representations, we know that it is essential to constrain the representations for effective OOD detection. The auxiliary information provides additional supervision to better constrain the representations for ID. Also, the auxiliary information is aligned with the activation-based representations to perform both detecting and generalizing to OOD in the same representation space. In the following section, we compare the activation-based representations learned with auxiliary information with gradient-based representations from a geometric perspective to understand their advantages.

5.2 Geometric Interpretation of Activation-based Representation Learned with Auxiliary Information

We geometrically interpret the aligned representations of the auxiliary information and visual data for OOD characterization. Assume that we have annotated visual data for ID and have class attribute data as auxiliary information for both ID and OOD. Although it is not feasible to have all annotated data for OOD, the attribute data for classes in OOD can be collected with significantly less amount of cost than annotated data. A vision encoder, f_v, and an attribute encoder, f_a, which consist of linear layers, output the visual representation, z_v, and the attribute representation, z_a, as follows:

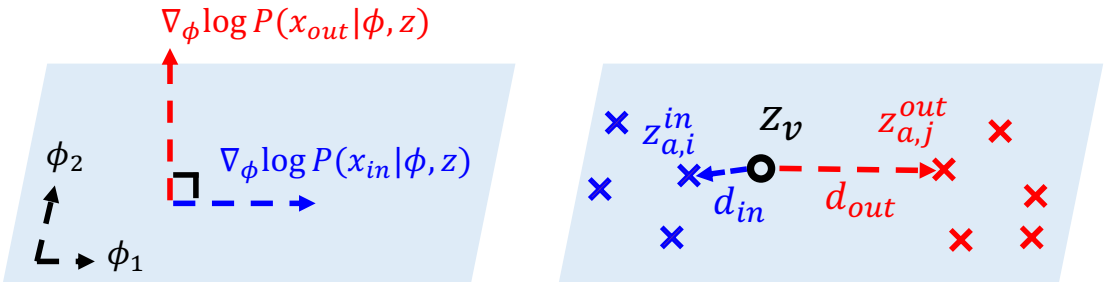
$$z_v = f_v(x), \tag{5.1}$$

$$z_a = f_a(a), \tag{5.2}$$

where x and a are visual and attribute data, respectively. Since we have attribute data for classes in ID and OOD, we denote the attribute representation for ID classes as z_a^{in} and that for OOD classes as z_a^{out}. To utilize the attribute data, the representations for the visual data and the attribute data need to be aligned. The alignment is obtained by training the network to minimize the distance between the representations for the ID visual data and its corresponding ID attribute data. Different distance metrics can be utilized depending on the applications and we assume the Euclidean distance for the simplicity of explanation.

We visualize OOD detection using the activation-based representations learned with the attribute data in Figure 5.1. Since attribute data is available for not only ID and but also OOD, we can use OOD attribute data as another reference signal to detect OOD. During testing, assume that the network is given a visual input, which is i-th class out of total N number of classes in a dataset. Assume that the i-th class is ID. This visual input and the attribute data for N classes are projected into the shared representation space. Because of the alignment enforced during training, the visual representation, z_v, is mapped closely to its corresponding ID attribute representation, $z_{a,i}^{in}$, while far away from OOD attribute representations, $z_{a,j}^{out}$, where $j \neq i$. Based on this alignment formed between visual and attribute representations, we can measure the distance from the visual representation to the ID attribute representation, d_{in}, and to the OOD attribute representation, d_{out}. These distances can be compared to characterize whether the visual input is closer to ID or OOD.

We highlight the complementary characteristics of activation-based representations learned with auxiliary information and gradient-based representations in Figure 5.1 (a) and (b). The advantage of the gradient-based representations in OOD detection is resulted from effectively constraining the ID representations. In particular, the gradients from ID data, $\nabla_\phi \log P(x_{in}|\phi, z)$, are constrained in the tangential plane of the manifold and distinguished from those of OOD data. In activation-based representations, the auxiliary in-



(a) Gradient-based representation

(b) Activate-based representation learned with auxiliary information

Figure 5.1: Comparison between gradient-based representations and the activation-based representations learned with auxiliary information.

formation plays a critical role to constrain the ID representation. The representations of attribute data is utilized as an anchor to constrain distance between the visual and the anchor representations. This constraint helps distinguishing the distance to ID attribute representations from the distance to OOD attribute representations. The limitation that gradient-based representations are not sufficiently effective for generalizing to OOD is also overcome by using activation-based representations. Since the activation-based representations characterize what the neural network know, it can more powerful representations for target tasks with OOD data. Details of learning activation-based representations with auxiliary information are discussed in the following section.

5.3 Representation Learning Using A Two-Stream Autoencoder

In this section, we start by defining notations for ID and OOD data and explain the training a two-stream autoencoder which learns activation-based representations with auxiliary information.

5.3.1 Problem Setup

We first define notations for the training data. Assume that $\mathcal{X}_{tr}^S, \mathcal{Y}^S, \mathcal{A}^S$ denote sets of visual features for ID training images, ID seen classes, and ID seen class attributes, respectively. Since there is an associated attribute for each class, the sets for seen classes

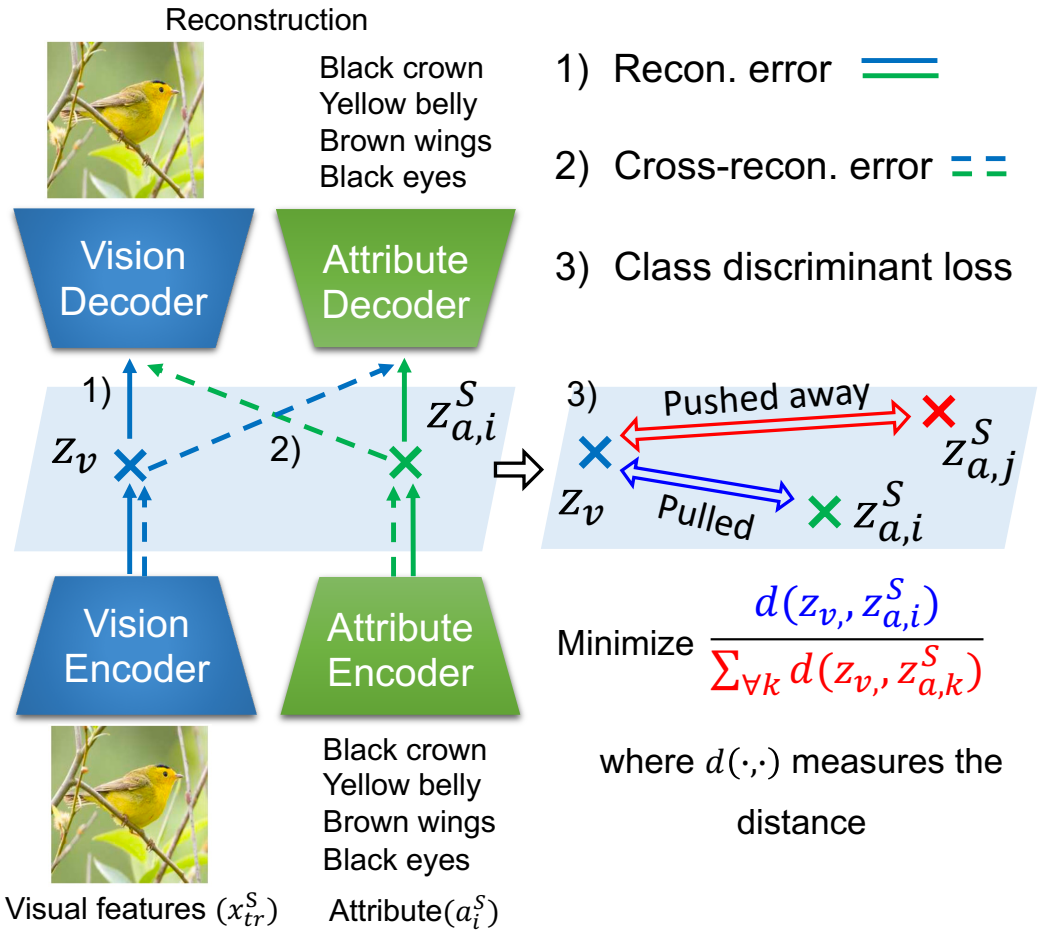


Figure 5.2: Training of the two-stream autoencoder.

and their attributes can be written as $\mathcal{Y}^S = \{y^S_1, \cdots, y^S_{|\mathcal{Y}^S|}\}$ and $\mathcal{A}^S = \{a^S_1, \cdots, a^S_{|\mathcal{Y}^S|}\}$, where $|\mathcal{Y}^S|$ defines the number of seen classes. If the class of a visual feature vector, x^S_{tr}, is y^S_i, the training sample can be given as a pair of (x^S_{tr}, a^S_i, y^S_i). We also have access to OOD unseen class attributes, $\mathcal{A}^U = \{a^U_1, \cdots, a^U_{|\mathcal{Y}^U|}\}$, and their associated unseen classes, $\mathcal{Y}^U = \{y^U_1, \cdots, y^U_{|\mathcal{Y}^U|}\}$, but do not have access to unseen class visual features during training. Assume that a set of visual features for seen class test images and that for unseen class test images are denoted as $\mathcal{X}^S_{te}$ and $\mathcal{X}^U_{te}$, respectively. The goal of OOD detection is to detect whether the visual feature, x_{te}, is from $\mathcal{X}^S_{te}$ or $\mathcal{X}^U_{te}$ using annotated seen visual features and all the attributes.

5.3.2 Two-stream Autoencoder

We use a two-stream autoencoder to learn representations that associate visual features with attributes. The two-stream autoencoder consists of a vision stream and an attribute stream. Also, each stream has an encoder and a decoder denoted as f_v and g_v, respectively for the vision stream and f_a and g_a for the attribute stream.

We train the autoencoder by imposing three different losses as shown in Figure 5.2. The first loss is a reconstruction error, $\mathcal{L}_{recon}$. Assume that a vision input, x_{tr}^S, the class of which is y_i and an associated attribute, a_i^S, are given to the autoencoder. Reconstruction for the vision and the attribute input can be denoted as $g_v(f_v(x_{tr}^S))$, $g_a(f_a(a_i^S))$, respectively. We measure the l_1 distance between the input and the reconstruction for each modality to obtain the reconstruction error. The reconstruction error for each modality is combined as follows:

$$\mathcal{L}_{recon} = \|x_{tr}^S - g_v(f_v(x_{tr}^S))\|_1 + \|a_i^S - g_a(f_a(a_i^S))\|_1. \tag{5.3}$$

In addition, we impose a cross-reconstruction error to align representations from visual features and attributes. The cross-reconstruction error has been widely used in the context of multimodal representation learning [113]. In particular, we train the autoencoder model to reconstruct one modality input from the other modality input as depicted in Figure 5.2 2). The visual features and the attributes are sequentially processed by the vision encoder and the attribute decoder, and attribute encoder and the vision decoder, respectively. We use l_1 distance to measure the cross-reconstruction error, $\mathcal{L}_{cross}$, which is formulated as follows:

$$\mathcal{L}_{cross} = \|x_{tr}^S - g_v(f_a(a_i^S))\|_1 + \|a_i^S - g_a(f_v(x_{tr}^S))\|_1. \tag{5.4}$$

Finally, we train the model with a cross entropy loss, $\mathcal{L}_{cls}$, to obtain class discriminant latent representations. The class discriminant representations are essential for the target

task such as image recognition to be easily performed. As shown in Figure 5.2 3), we first obtain the visual latent representation as $z_v = f_v(x_{tr}^S)$ and all the seen attribute latent representations as $z_{a,k}^S = f_a(a_k^S)$, where $k = 1, \ldots, |\mathcal{Y}^S|$. When the class of visual input is assumed to be y_i, the loss is computed as

$$\mathcal{L}_{cls} = -\log\left(\frac{\exp(-\|z_v - z_i^S\|_2)}{\sum_{k=1}^{|\mathcal{Y}^S|}\exp(-\|z_v - z_k^S\|_2)}\right). \tag{5.5}$$

The term in the numerator contributes to minimize the distance for the positive pair of visual and attribute representations while the terms in the denominator enforce to maximize the distance for negative pairs.

The overall loss for the autoencoder, $\mathcal{L}_{all}$ is given as follows:

$$\mathcal{L}_{all} = \mathcal{L}_{recon} + \mathcal{L}_{cross} + \alpha\mathcal{L}_{cls}, \tag{5.6}$$

where α is empirically determined to balance the cross entropy loss and the reconstruction losses. Aforementioned losses are also commonly explored in other existing works [84, 114]. However, they mainly focus on aligning the visual and attributes representations while the OOD detection capability of the representations are not explored. We highlight that the our main contribution is on developing the OOD detection algorithm using these generic aligned representations, which is discussed in the next section.

5.4 Unseen Class Detection and Classification

We primarily focus on obtaining descriptive features that can characterize unseen OOD classes from the two-stream autoencoder. In particular, we use the distance between visual and attribute representations as a feature for unseen class detection. The attributes that describe the seen and the unseen classes are available during training and testing. Therefore, we use the attribute representations of the autoencoder as references and compute the distance from the visual representation to the seen and unseen attribute representations.

$$r_{latent} = d^{S}_{latent} / d^{U}_{latent}$$

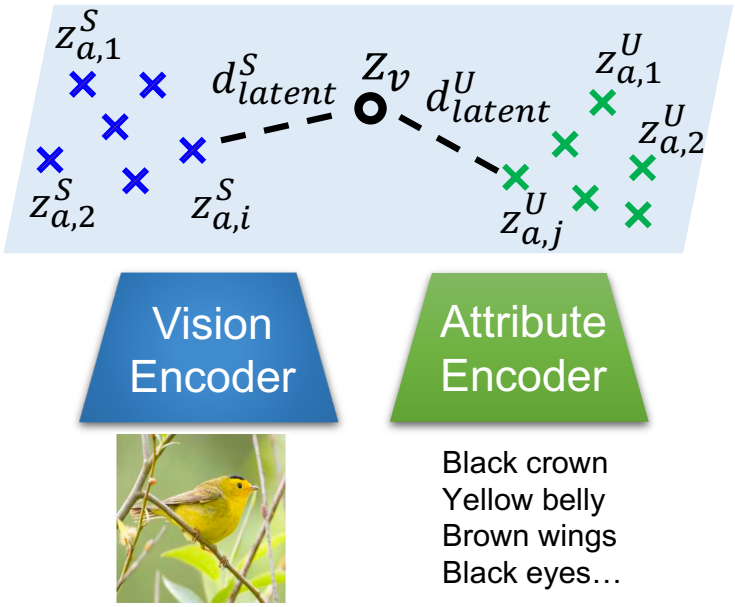


Figure 5.3: Unseen class detection using distance features in the latent space of the two-stream autoencoder.

Since the autoencoder is trained to align seen class visual input and its attribute, the seen class visual input will reside closer to seen attribute than unseen class visual input in the representation spaces. Also, given that the autoencoder is trained to learn the association between visual input and its relevant attribute, we hypothesize that the unseen class visual input will stay closer to unseen attribute than seen class visual input. Hence, by comparing the distance between visual and attribute representations, we can predict whether the query is a seen or an unseen class.

We obtain the distance features in both latent space and cross-reconstruction space of the two-stream autoencoder. In particular, the latent space is relatively lower dimension than cross-reconstruction space in our two-stream autoencoder. Therefore, distances obtained in those two spaces abstract features at different semantic levels. We use both low and high dimensional distance features to define the unseen class score which indicates the

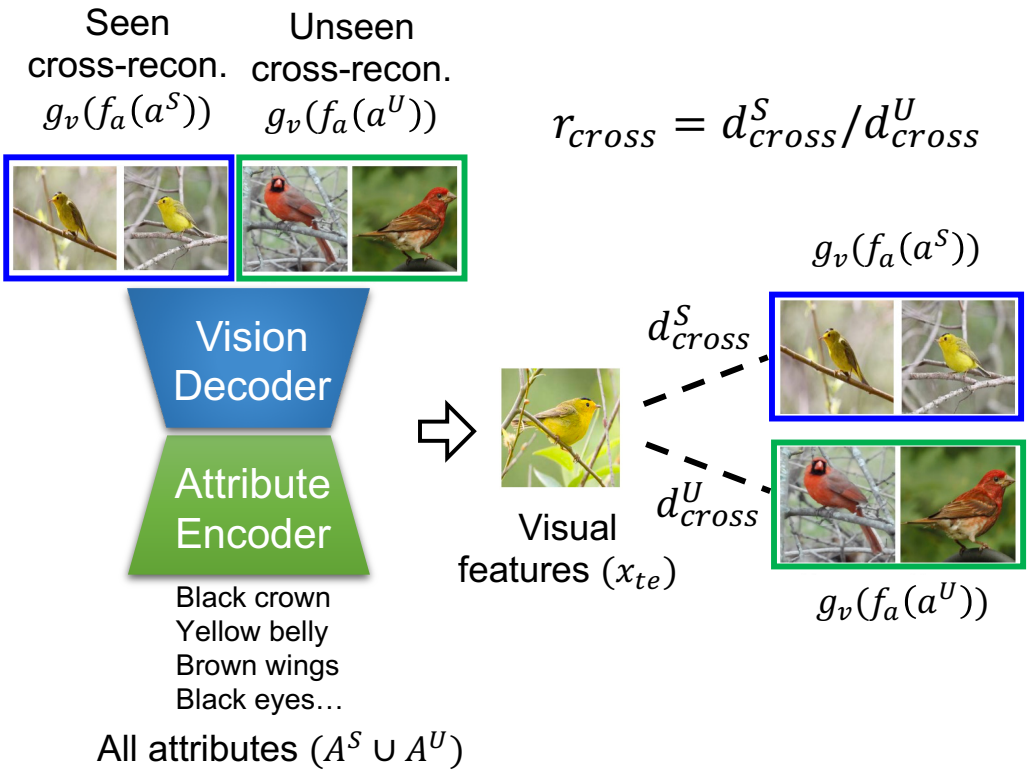


Figure 5.4: Unseen class detection using distance features in the cross-reconstruction space of the two-stream autoencoder.

possibility of the query sample being an unseen class. The detailed steps for unseen class score calculation in both spaces and the final classification are discussed in the following subsections.

5.4.1 Unseen class detection in the latent space

We visualize unseen class detection using the latent representations in Figure 5.3. The latent representation of the query visual feature, x_{te}, is obtained as $z_v = f_v(x_{te})$. We also generate the latent representations for all the seen and unseen attributes in $\mathcal{A}^S \cup \mathcal{A}^U$. We denote all the seen and unseen attribute latent representations as $\mathcal{Z}^S_a = \{z^S_{a,1} \dots z^S_{a,|\mathcal{Y}^S|}\}$ and $\mathcal{Z}^U_a = \{z^U_{a,1} \dots z^U_{a,|\mathcal{Y}^U|}\}$, respectively. We extract distance features by computing the minimum distance from the visual representation to the seen and the unseen attribute representations. The distance features for the seen class, d^S_{latent}, and the unseen class, d^U_{latent},

are calculated as follows:

$$d^S_{latent} = \min_i \exp(\|z_v - z^S_{a,i}\|_2) \tag{5.7}$$

$$d^U_{latent} = \min_j \exp(\|z_v - z^U_{a,j}\|_2) \tag{5.8}$$

We use the exponential of l_2 distance since this term is used in the cross entropy loss imposed in the latent space. We utilize the ratio between these distances to obtain an unseen class score in the latent space, r_{latent}, which is defined as $r_{latent} = d^S_{latent}/d^U_{latent}$. The seen class visual input will result in smaller d^S_{latent}, larger d^U_{latent}, and consequently smaller r_{latent} than the unseen class visual input. Therefore, high r_{latent} indicates that the query input is likely to be an unseen class. We can detect the query as an unseen class when the unseen class score is above a certain threshold. Otherwise, the query is detected as a seen class.

5.4.2 Unseen class detection in the cross-reconstruction space

We can also obtain the unseen class score in the cross-reconstruction space as shown in the right side of Figure 5.4. Similar to the calculation of the unseen class score in the latent space, we input all the seen and unseen attributes to the trained attribute encoder. Then, we use the vision decoder to cross-reconstruct images from attributes. The cross-reconstruction of seen and unseen class attributes are denoted as $\{g_v(f_a(a^S_1)) \dots g_v(f_a(a^S_{|\mathcal{Y}^S|}))\}$ and $\{g_v(f_a(a^U_1)) \dots g_v(f_a(a^U_{|\mathcal{Y}^U|}))\}$, respectively. We extract the distance features in the cross-reconstruction space by comparing the query visual features and the cross-reconstructions from attributes. The minimum distance from the query visual input to the seen cross-reconstruction, d^S_{cross}, and to the unseen cross-reconstruction, d^U_{cross}, are computed as fol-

lows:

$$d^S_{cross} = \min_i \|x_{te} - g_v(f_a(a^S_i))\|_1 \tag{5.9}$$

$$d^U_{cross} = \min_j \|x_{te} - g_v(f_a(a^U_j))\|_1. \tag{5.10}$$

l_1 distance is used as the cross-reconstruction error imposed during training. We combine two distance features by computing the ratio $r_{cross} = d^S_{cross}/d^U_{cross}$ and utilize it as an unseen class score from the cross-reconstruction space. When the query visual input is from seen classes, the input should be close to one of seen class cross-reconstructions and achieve smaller d^S_{cross} than the unseen class input. Also, since the autoencoder is trained to cross-reconstruct images that are relevant to the given attributes, the unseen class input will be similar to one of the unseen cross-reconstructions and result in lower d^U_{cross} than the seen class input. Therefore, we can detect unseen classed by comparing r_{cross}. We first use r_{latent} and r_{cross} individually to examine the OOD detection capability of activation-based representations learned with auxiliary information.

5.5 Experiments

We validate the effectiveness of the proposed OOD detection method through rigorous baseline experiments. In particular, we compare the three different representations which are activation-based representations, gradient-based representations, and activation-based representations learned with auxiliary information for OOD detection. These experiments show that the activation-based representations learned with auxiliary information successfully complement both activation and gradient, and achieve improved performance for OOD detection.

Figure 5.5: Sample images from CUB, SUN, AWA1, and AWA2.

5.5.1 Experimental Setup

We validate the proposed representations using four benchmark image recognition datasets: Caltech-UCSD Birds-200-2011 (CUB) [115], SUN Attribute (SUN) [116], Animals with Attribute 2 (AWA2) [117], and Animals with Attribute 1 (AWA1) [65]. Also, we use the proposed splits in [117] for all the datasets. CUB is a fine-grained dataset with $11,788$ bird images from 200 species. For the attributes, we use text representations obtained by averaging 10 sentence features per image [118]. SUN is also a fine-grained image dataset which contains $14,340$ visual scene images from 717 classes. Each scene class is annotated with a 102-dimensional attribute representation. AWA2 and AWA1 are both coarse-scale image datasets. AWA2 and AWA1 consist of $37,322$ and $30,475$ animal images, respectively. Both datasets have 50 classes and each class is annotated with a 85-dimensional attribute representation. As suggested in [117], we use 2048-dimensional image representations obtained from the top-layer pooling units of ResNet-101 [12] pre-trained on ImageNet [119] as visual input for all four datasets. Sample images from four datasets are visualized in Figure 5.5. We use 150, 645, 40, and 40 classes in CUB, SUN, AWA2, and AWA1 for ID and remaining 50, 72, 10, 10 classes for OOD in this experiment. We note that more fine-grained and complicated images such as bird species or scenes are utilized in comparison with simple digit image or object image datasets utilized in the previous chapters. Also, we do not use only one class as ID while more classes are added in the ID to make the

Table 5.1: AUROC performance of OOD detection based on different representations.

OOD Detection Score	Datasets			
	CUB	SUN	AWA2	AWA1
Recon	0.4658	0.5403	0.6890	0.6798
Grad	0.4999	0.4998	0.5000	0.5000
Attribute (r_{latent})	**0.8415**	**0.7740**	**0.9337**	**0.9168**
Attribute (r_{cross})	0.8078	0.7685	0.9330	0.9073

detection problem more challenging and feasible for real-world scenarios.

We train three different autoencoders using only ID data to obtain representations for OOD detection. First, we train a vision autoencoder with the reconstruction error which is defined as the first term of Equation 5.3.2. The reconstruction error is used as an OOD detection score. Second, we train the vision autoencoder with the gradient constraint as Equation 4.2. The gradient loss is used to detect OOD. Finally, we train a two-stream autoencoder using the loss defined in Equation 5.3.2. We use r_{latent} and r_{cross} separately as OOD scores. We compare the OOD detection performance of three different representations in terms of AUROC. The encoder and the decoder in the autoencoder consist of two linear layers and ReLUs are used after the first layer of the encoder and the decoder. The dimension of the latent space is 64 and the batch size of 64 is used. We use Adam optimizer [106] with the learning rate of 1.5×10^{-4} to train all the autoencoders for 100 epochs.

5.5.2 Results

We report the AUROC performance for OOD detection in Table 5.1. `Recon`, `Grad`, `Attribute` (r_{latent}), `Attribute` (r_{cross}) denote the reconstruction error, the gradient loss, r_{latent}, and r_{cross} from the aligned visual and attribute representations respectively. The highest score in each dataset is highlighted in bold. r_{latent} from the activation-based representation learned with auxiliary information consistently achieves the best performance across four datasets. `Recon` achieves higher performance in coarse-grained AWA2 and AWA1 datasets than fine-grained CUB and SUN dataset. In particular, the AU-

ROC score in `Recon` is lower than 0.5, which clearly shows its limitation in fine-grained datasets. `Grad` achieves around 0.5 for all datasets. We observe that when ID contains images from several classes, constraining gradients from different classes becomes extremely challenging. To differentiate representations of images from different classes, the neural networks need to diverging gradients for those classes. This leads gradients to be not sufficiently constrained through the directional constraint and causes poor OOD detection performance. On the other hand, `Attribute` utilizes the auxiliary information to provide effective supervision to obtain high AUROC scores. This shows that aligned representations of visual and attribute data can be utilized for OOD detection in real-world scenarios where images with diverse granularity and classes are included in ID. While `Attribute` (r_{cross}) achieves lower AUROC scores than `Attribute` (r_{latent}), it significantly outperforms `Recon` and `Grad`. Since r_{cross} and r_{latent} are obtained from two different representations of the two-stream autoencoder, they can complement each other to further improve the OOD detection performance. This complementary features from r_{cross} and r_{latent} are more rigorously discussed in the next chapter. In Figure 5.6, we show the ROC curves which visualize the OOD detection performance of `Recon`, `Grad`, and `Attribute` (r_{latent}). The curve which is closer to top right corner indicates the better performance. From these curves, we validate that `Attribute` (r_{latent}) achieves the highest true positive rate with the lowest false positive rate.

5.6 Summary

In this chapter, we propose learning aligned activation-based representations for visual and attribute data to perform OOD detection. We first thoroughly examine the current limitation of activation-based and gradient-based representations. In particular, the activation-based and the gradient-based representations face challenges for OOD detection when ID contains diverse granularity and classes of data. To overcome the limitations, we use attribute data as another supervision to constrain the activation-based representations. Two-stream

autoencoder is trained to learn the aligned representations between visual and attribute data. In addition, we characterize OOD by measuring the distance from visual representation to seen and unseen class attribute representations. From our controlled experiments, we validate that the activation-based representations learned with auxiliary information outperform both activation-based and gradient-based representations in terms of OOD detection performance. Since the representations effectively characterize OOD even when ID data becomes diverse and complicated, the activation-based representations learned with auxiliary information enable the generalization to OOD.

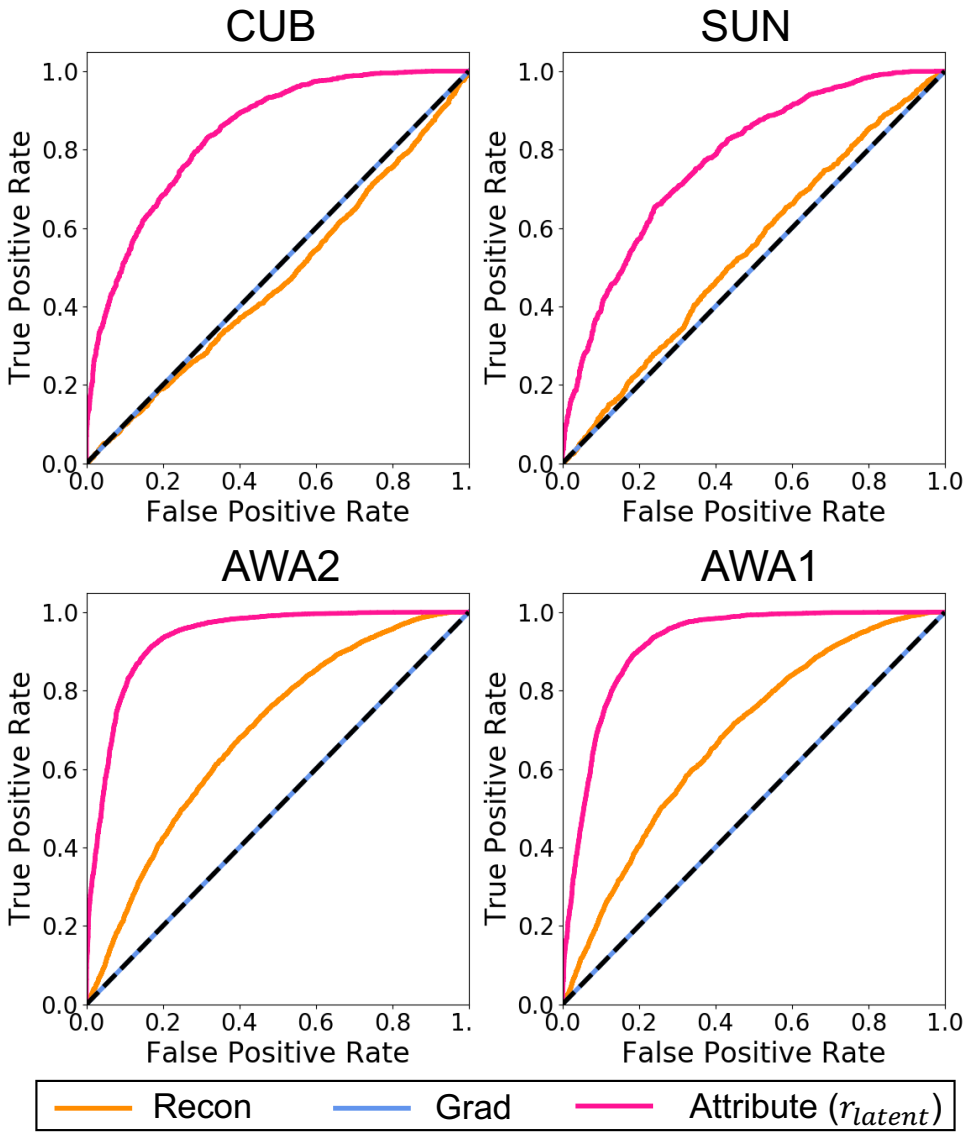


Figure 5.6: ROC curves for the OOD detection using different representations.

CHAPTER 6

GENERALIZATION TO OUT-OF-DISTRIBUTION

The success of generalization to OOD relies on calibrating bias toward ID in neural networks. In this chapter, we focus on utilizing representations that we develop and introduce in the previous chapters to calibrate the bias and solve target tasks for OOD data. The activation-based representations learned with auxiliary information are particularly used to develop a gating model which characterizes OOD data and separates them out from ID. This gating model mitigates the bias issue in neural networks and enables them to perform generalization to OOD. We validate the gating model in the application of generalized zero-shot learning (GZSL). Through rigorous baseline experiments and ablation study, we highlight the advantages of the gating model. In addition, we achieve state-of-the-art performance in four benchmark image recognition datasets with at least 20% less number of model parameters than state-of-the-art methods relying on generative models.

In summary, the main contributions of this paper are three folds:

i We propose a two-stream autoencoder-based gating model which prevents biased prediction toward seen classes and achieve state-of-the-art performance in four benchmark image recognition datasets.

ii We rigorously explore the applicability of the proposed method as a standalone and to existing state-of-the-art methods.

iii We conduct diverse ablation studies to validate the gating performance and the computational efficiency of the proposed gating model.

6.1 Motivation and Challenges for Generalization to Out-of-Distribution

Advancement in machine learning has primarily been driven by a large amount of labeled data. In particular, a supervised learning framework which utilizes fully annotated data such as ImageNet [119] achieves state-of-the-art performance in diverse applications such as object recognition, detection, and segmentation [12, 13, 14]. However, supervised learning has clear limitations when generalizing in numerous real-world scenarios because of expensive data collection and annotation. Also, to generalize the supervised model to a new class, the model needs to be trained with a large amount of data for the new class even though the new class is similar to other trained classes. These limitations motivate the development of other learning paradigms that do not require fully annotated data.

Generalization to OOD can be achieved by learning representations with auxiliary information such as attributes of both seen and unseen class. For example, in the application of image recognition, assume that a classifier is trained for 'horse' class and 'striped cat' class. If we have auxiliary information of textual description for a new class 'zebra' such as "zebra is a horse with stripes", the classifier can associate the 'horse' features and 'stripe' features from training images to learn the new class 'zebra'. Hence, when both visual and attribute data are aligned in the representation space, the attribute can bridge between ID and OOD and transfer knowledge learned from ID to perform the target tasks in OOD.

The aligned representations of visual and attribute data are still required to overcome a challenge for the successful generalization to OOD. The challenge is a biased model prediction caused by the inherently unbalanced training set. During the training of neural networks in ID, both visual and attribute features are available for seen classes while only attribute features are provided for unseen classes. Hence, the unbalanced training set causes models to overfit on ID seen class data and perform well for seen classes but poorly for OOD unseen classes. Several approaches [89, 90] have been proposed to overcome this challenge by calibrating prediction scores for seen classes. However, we still observe that

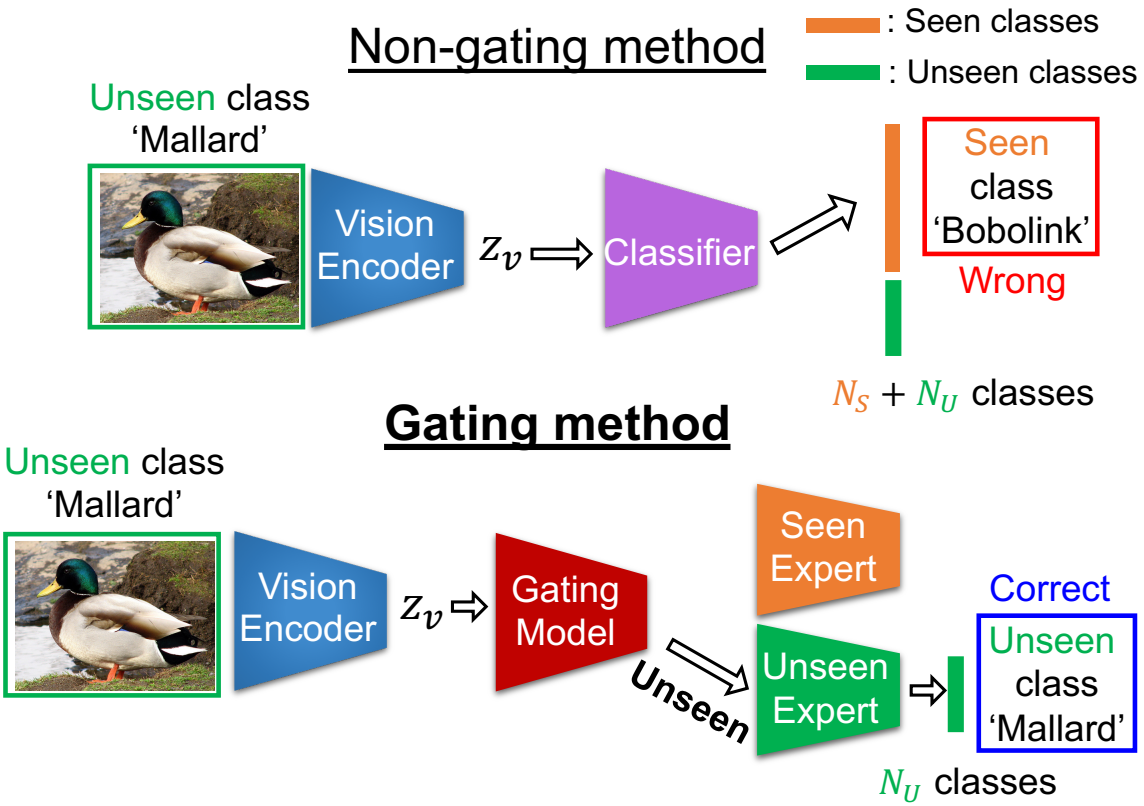


Figure 6.1: Comparison between the non-gating method and the gating method.

most of the unseen classes are misclassified as seen classes. In these calibration methods, the classifier makes a prediction out of the search space that contains both seen and unseen classes. Thus, the prediction scores for unseen classes cannot completely avoid competing with the biased prediction scores for seen classes. We propose using a gating model to tackle the biased prediction challenge in generalization to OOD.

6.2 Gating Model for OOD Detection and Classification

In Figure 6.1, we compare the standard (non-gating) method and the gating method for generalization to OOD. For both models, a visual representation, z_v, is obtained by giving an input image to the vision encoder. Assume that an unseen class image is given to the models. N_S and N_U denote the number of seen classes in ID and unseen classes in OOD. The gating method consists of three components, which are a gating model, a seen expert,

and an unseen expert. The seen expert and the unseen expert are trained to correctly classify seen and unseen classes, respectively. The gating model first performs unseen class (OOD) detection which aims at correctly predicting whether the image is from the seen or the unseen classes. Based on the unseen class detection result, either the seen expert or the unseen expert is chosen to predict the class. While in the standard non-gating method, a class is predicted out of total $N_S + N_U$ classes, the gating method predicts a class out of either N_S classes or N_U classes. Thus, the model can avoid comparing the biased seen class prediction scores with the unseen class prediction scores in the gating method.

We propose a two-stream autoencoder-based gating model which possesses several advantages over other methods generalizing to OOD. In particular, we utilize representations from latent space and cross-reconstruction space to characterize association between query visual input and attributes and perform accurate unseen class detection. Also, our two-stream autoencoder provides a unified framework for both the gating model and the unseen expert. The latent representations are trained to be class-discriminant and directly utilized for unseen class classification. Therefore, no additional unseen expert needs to be trained, which leads to the computational efficiency of the proposed method. Furthermore, we show that both experts can be separately optimized and the gating model can be easily combined with other state-of-the-art methods. We validate the proposed approach in the application of generalized zero-shot learning for image recognition.

6.2.1 Problem Setup

We first define notations for the training data. Assume that $\mathcal{X}^S_{tr}, \mathcal{Y}^S, \mathcal{A}^S$ denote sets of visual features for seen class training images, seen classes, and seen class attributes, respectively. Since there is an associated attribute for each class, the sets for seen classes and their attributes can be written as $\mathcal{Y}^S = \{y^S_1, \cdots, y^S_{|\mathcal{Y}^S|}\}$ and $\mathcal{A}^S = \{a^S_1, \cdots, a^S_{|\mathcal{Y}^S|}\}$, where $|\mathcal{Y}^S|$ defines the number of seen classes. If the class of a visual feature vector, x^S_{tr}, is y^S_i, the training sample can be given as a pair of (x^S_{tr}, a^S_i, y^S_i). We also have ac-

cess to unseen class attributes, $\mathcal{A}^U = \{a_1^U, \cdots, a_{|\mathcal{Y}^U|}^U\}$, and their associated unseen classes, $\mathcal{Y}^U = \{y_1^U, \cdots, y_{|\mathcal{Y}^U|}^U\}$, but do not have access to unseen class visual features during training. Assume that a set of visual features for seen class test images and that for unseen class test images are denoted as $\mathcal{X}_{te}^S$ and $\mathcal{X}_{te}^U$, respectively.

There exist two different evaluation frameworks for generalization to OOD, which are standard zero-shot learning (ZSL) and generalized zero-shot learning (GZSL). In standard ZSL, test images are drawn only from unseen classes. Hence, the goal is to learn a classifier, f, which can predict the correct label for x_{te} and it can be formulated as $f(x_{te}, a; \theta_f) : (\mathcal{X}_{te}^S \cup \mathcal{X}_{te}^U) \times (\mathcal{A}^S \cup \mathcal{A}^U) \rightarrow (\mathcal{Y}^S)$, where θ_f is the model parameters. However, in GZSL, a visual feature vector, x_{te}, are drawn from the union of seen and unseen class test sets, $\mathcal{X}_{te}^S \cup \mathcal{X}_{te}^U$. Therefore, the goal of learning a classifier for GZSL is can be formulated as $f(x_{te}, a; \theta_f) : (\mathcal{X}_{te}^S \cup \mathcal{X}_{te}^U) \times (\mathcal{A}^S \cup \mathcal{A}^U) \rightarrow (\mathcal{Y}^S \cup \mathcal{Y}^U)$, where θ_f is the model parameters. GZSL focuses on achieving high accuracy for both seen and unseen class test images, which is more challenging than ZSL. In this dissertation, we specifically tackle the problem of GZSL.

6.2.2 OOD Detection and Classification

The success of the gating model approach heavily relies on the OOD detection performance since the input data needs to be correctly assigned to each expert. Therefore, we focus on improving the distance features obtained from the representation learned with attribute data by the two-stream autoencoder. The two-stream autoencoder is trained with the reconstruction error, $\mathcal{L}_{recon}$, the cross-reconstruction error, $\mathcal{L}_{cross}$, and the cross entropy loss, $\mathcal{L}_{cls}$, as in Equation 5.3.2. From the autoencoder, we obtain distance features in the latent space, r_{latent}, and the cross-reconstruction space, r_{cross}, for OOD detection. We finally combine both distance features to complement each other and improve the unseen class detection

performance. We obtain the final unseen class score, r_{all}, as

$$r_{all} = \frac{\min_i \|x_{te} - g_v(f_a(a_i^S))\|_1 + \beta \exp(\|z_v - z_{a,i}^S\|_2)}{\min_j \|x_{te} - g_v(f_a(a_j^U))\|_1 + \beta \exp(\|z_v - z_{a,j}^U\|_2)}, \tag{6.1}$$

where β is a hyperparameter to balance two distances from the latent space and the cross-reconstruction space. We perform baseline experiments to compare the GZSL performance with three different unseen class scores, r_{latent}, r_{cross}, r_{all}. We use r_{all} which shows the best performance in the baseline experiments for the state-of-the-art comparison.

The seen and unseen expert can be independently trained or obtained as a byproduct from the autoencoder. For the seen expert, we train a supervised linear classifier with one layer, f_{cls}^S, using available visual features of seen class training images. For the unseen expert, we do not train any additional model but performs 1-nearest neighbor classification in the latent space to predict the class. To be specific, we measure the l_2 distance between visual latent representation and all the unseen attribute latent representations, and the class of the closest unseen attribute representation is predicted as a label. We can formulate the overall seen and unseen class detection, and the classification as follows:

$$\hat{y} = \begin{cases} f_{cls}^S(x_{te}) & \text{if} \quad r_{all} < \tau, \\ y_k^U \quad \text{where} \quad k = \arg\min_j \|z_v - z_{a,j}^U\|_2 & \text{else}, \end{cases} \tag{6.2}$$

where $\hat{y}$ is the final class prediction of x_{te}. The hyperparameters β and τ are found using the validation set provided in [117]. Also, by following the training protocol described in [87], we re-train the model from the scratch using the union of the training and the validation sets after finding β and τ. We note that these seen and unseen experts are one of the simplest models for classification. By showing that the gating model achieves the state-of-the-art performance even with these shallow models, we highlight the contribution of bias calibration from the gating model for generalization to OOD. Since we utilize a compact model of the two-stream autoencoder for the gating, the proposed method is called

GatingAE.

6.2.3 Advantages of the proposed method

GatingAE has several advantages over other GZSL algorithms in terms of prediction performance, applicability to other state-of-the-art algorithms, and computational efficiency. We highlight these advantages in details below.

First, GatingAE enables to separate the prediction search space for the seen and the unseen expert, which leads to significant gain in the overall GZSL performance. Non-gating models and soft-gating models such as [87] predicts a class out of total $|\mathcal{Y}^S| + |\mathcal{Y}^U|$ number of classes. However, GatingAE separates the search space and the experts predict either seen classes or unseen classes. In this setup, the biased prediction scores for seen classes are not directly compared with the prediction scores for unseen classes to make the final prediction. Thus, we can mitigate the effect of the bias toward seen classes in GZSL. Also, the dimension of the search space is reduced from $|\mathcal{Y}^S| + |\mathcal{Y}^U|$ to $|\mathcal{Y}^S|$ and $|\mathcal{Y}^U|$ for the seen and the unseen expert, respectively. The reduced dimension of the search space allows experts to focus on less number of classes for the classification, which lead to better accuracy performance. (subsection 6.3.2, Table 6.1)

Second, each expert can be independently optimized for improving the classification performance in GatingAE. GatingAE decomposes the entire framework of GZSL into three components: an autoencoder for gating, a seen expert, and an unseen expert. Since experts play independent roles in the pipeline, each of them can be optimized separately to better achieve its own goal. For instance, the seen expert, f^S_{cls}, in GatingAE is not affected by other tasks such as learning the association between visual features and attributes during training. Therefore, the seen expert can focus only on improving the performance for seen classes and consequently contributes to achieve better GZSL performance. (subsection 6.3.2, Table 6.1)

Third, GatingAE can be easily combined with other existing state-of-the-art methods

to further improve the performance. Since each expert can be separately improved in GatingAE, state-of-the-art methods can be simply utilize as a seen or an unseen expert. Even when the state-of-the-art-method combined with GatingAE does not achieves better GZSL performance than GatingAE, our proposed gating method can still benefit from it. For instance, we show that we can finetune GatingAE with the generated data from f-CLSWGAN [75] and achieve improved performance over both base GatingAE and f-CLSWGAN. (subsection 6.3.3, Table 6.3)

Fourth, GatingAE comprehensively characterizes unseen classes at different levels of data abstraction. In our autoencoder implementation, we set the latent dimension lower than the original input. Hence, the distance features obtained in the latent space and the cross-reconstruction space characterize unseen class samples in two different dimensional representation of the same data. Also, the low and high dimensional distance features complement each other and combined features provide comprehensive characterization of unseen classes. Accurate detection performance from complementary distance features ensures to choose a right expert for the the query data. (subsection 6.3.2, Table 6.1, subsection 6.3.4, Table 6.2, Table 6.4)

Finally, GatingAE is computationally efficient. One of the main approaches in GZSL is to use generative models such as GANs [18] and VAEs [17] to generate unseen visual features. GatingAE does not utilize these generative models which usually require a large number of parameters to train. In addition, GatingAE has a unified framework for the gating model and the unseen expert. For instance, the state-of-the-art soft-gating model [87] consists of a separate gating model and an unseen expert. However, in GatingAE, we use the autoencoder not only as a gating model but also for an unseen expert by performing 1-nearest neighbor classification in the latent space. Therefore, GatingAE requires significantly less number of model parameters and computational resources compared to other state-of-the art methods. (subsection 6.3.4, Table 6.5)

6.3 Experiments

We validate the effectiveness of the proposed gating model through rigorous baseline experiments. Also, we highlight the GZSL performance of GatingAE in comparison with other state-of-the-art methods. Finally, comprehensive ablation studies are conducted to experimentally support the advantages of GatingAE.

6.3.1 Experimental Setup

We validate the proposed representations using the same four benchmark image recognition datasets introduced in section 5.5, which are Caltech-UCSD Birds-200-2011 (CUB) [115], SUN Attribute (SUN) [116], Animals with Attribute 2 (AWA2) [117], and Animals with Attribute 1 (AWA1) [65]. We use the proposed splits in [117] for all the datasets. In particular, we use 150, 645, 40, and 40 classes in CUB, SUN, AWA2, and AWA1, respectively, as seen classes for the neural network. Remaining 50, 72, 10, 10 classes in CUB, SUN, AWA2, and AWA1, respectively, are utilized as unseen classes. We use average per-class top-1 accuracy which is a widely accepted evaluation metric for GZSL to evaluate the proposed method. In particular, we separately calculate the average accuracy for seen classes and unseen classes. We also report the harmonic mean (H) of the seen class accuracy (S) and the unseen class accuracy (U), which is calculated as $H = 2 \times U \times S/(U + S)$. For the evaluation of the unseen class detection performance, we use area under receiver operation characteristic curve (AUC) and false positive rate at true positive rate 0.95 (FPR).

The encoder and the decoder of each stream in the autoencoder consist of two linear layers and ReLUs are used after the first layer of the encoder and the decoder. The dimension of the latent space is 64 and the batch size of 64 is used. We use Adam optimizer [106] with the learning rate of 1.5×10^{-4} and train the two-stream autoencoder for 100 epochs. For the seen expert, we train the one layer linear classifier using the batch size of 32 and Adam optimizer with the learning rate of 0.001.

Table 6.1: Baseline comparison in CUB, SUN, AWA2, and AWA1 datasets. S: Seen class accuracy, U: Unseen class accuracy, H: Harmonic mean accuracy. Top 2 harmonic mean accuracies for each dataset are highlighted in bold.

Model	Seen Exeprt	CUB			SUN			AWA2			AWA1		
		S	U	H	S	U	H	S	U	H	S	U	H
No gating	1-NN	64.4	36.6	46.8	35.0	19.0	24.7	87.8	25.9	40.0	85.3	23.8	37.2
GatingAE	1-NN	55.0	54.2	54.6	29.8	48.1	36.8	81.0	54.7	65.3	76.8	54.9	64.0
(r_{latent})	Linear CLF	58.6	54.2	**56.4**	35.7	48.1	**40.9**	83.1	54.7	**66.0**	78.7	54.9	**64.7**
GatingAE	1-NN	45.0	58.8	51.0	27.4	50.4	35.5	79.0	55.4	65.1	73.1	55.8	63.3
(r_{cross})	Linear CLF	47.1	58.8	52.3	32.4	50.4	39.5	80.9	55.4	65.8	74.9	55.8	63.9
GatingAE (r_{all})	Linear CLF	58.1	54.9	**56.4**	38.1	45.4	**41.4**	81.3	57.3	**67.2**	72.8	59.7	**65.6**

Table 6.2: Gating performance comparison between GatingAEs and gating models proposed in COSMO. Ideally, higher harmonic mean accuracy (H), higher AUC, and lower false positive rate at true positive rate 0.95 (FPR) are desired. Top 2 scores in each evaluation metric are highlighted.

Gating Model	CUB			SUN			AWA1		
	H(↑)	AUC(↑)	FPR(↓)	H(↑)	AUC(↑)	FPR(↓)	H(↑)	AUC(↑)	FPR(↓)
MAX-SOFTMAX-3 [87]	43.6	0.734	0.796	38.4	0.610	0.923	53.1	0.886	0.568
CB-GATING-3 [87]	44.7	0.820	0.720	**40.1**	**0.777**	**0.775**	56.8	**0.925**	**0.455**
GatingAE (r_{latent})	**75.7**	**0.972**	**0.143**	38.0	**0.777**	**0.775**	**62.1**	0.889	0.566
GatingAE (r_{cross})	61.7	0.926	0.324	34.8	0.753	0.820	61.3	0.890	0.561
GatingAE (r_{all})	**74.9**	**0.970**	**0.156**	**38.8**	**0.779**	**0.762**	**62.7**	**0.894**	**0.550**

6.3.2 Baseline Comparison

We validate the effectiveness of the gating model through comprehensive baseline experiments in Table 6.1. We compare the GZSL performance of four different models. All four models are based on the same two-stream autoencoder trained as described in Section subsection 5.3.2. However, the gating approach used in the inference stage differs for them. As shown in the first column of Table 6.1, the first model (*No gating*) predicts the class through 1-nearest neighbor (1-NN) classification based on the latent representations of the autoencoder without any gating approaches. To be specific, the predicted class is given as $\hat{y} = y_k^{S+U}$, where $k = \arg\min_j \|z_v - z_{a,j}^{S+U}\|_2$, $y_k^{S+U} \in \mathcal{Y}^S \cup \mathcal{Y}^U$, $z_j^{S+U} \in \mathcal{Z}_a^S \cup \mathcal{Z}_a^U$. Since no gating approach is used, we note that the classification is made out of $|\mathcal{Y}^S| + |\mathcal{Y}^U|$ classes. The second and the third models use unseen class scores in the latent space (GatingAE (r_{latent})), and in the cross reconstruction space, (GatingAE (r_{cross})), for gating, respectively. For the two models, we use both 1-NN classifier and a linear classifier (Linear CLF) as seen experts and compare the performance. Finally, we combine distance features obtained in the latent space and the cross-reconstruction space, and use r_{all} as an unseen class score for gating. We use a linear classifier as a seen expert. For all the gating models, 1-NN classifier applied on the latent representations is used as an unseen expert. We report average per-class top-1 accuracy for seen classes (S), unseen classes (U), and the harmonic mean of them (H), in CUB, SUN, AWA2, and AWA1 datasets.

Effectiveness of the proposed gating model (No gating vs. GatingAE) We highlight the contribution of the gating model by comparing the performance of the *No gating* model and GatingAE models using r_{latent} and r_{cross} separately as unseen class scores. For fair comparison, we compare models using the 1-NN classifier as seen experts. GatingAE (r_{latent}) with the 1-NN classifier significantly outperforms the *No gating* model by $7.8, 12.1, 25.3,$ and 26.8 in terms of the harmonic mean accuracy in CUB, SUN, AWA2, and AWA1, respectively. Furthermore, GatingAE (r_{cross}) with the 1-NN classifier shows higher harmonic mean accuracy than the No gating model by a margin of 4.2, 10.8, 25.1,

and 26.1 in the four datasets.

We believe two advantages of GatingAE mainly contribute to the significantly improved performance. First, GatingAE prevents the biased model prediction toward seen classes. Because of the visual data only available for seen classes during training, the classifier overfits to seen classes. In GatingAE, the gating model separates the prediction search space and the class is predicted only among seen classes or unseen classes for each sample. This prevents unseen class prediction scores from being directly compared with the biased seen class scores. On the other hand, in the *No gating* model, the biased seen class scores and the unseen class scores are directly compared and the class with the maximum score is predicted. This lead to misclassification of unseen class samples into seen classes. We observe that the seen accuracy of the *No gating* model is at least 1.76 times and at most 3.58 times higher than the unseen accuracy of the same model across the four datasets. GatingAE avoids this biased prediction and achieves significantly improved harmonic mean accuracy. Second, the gating approach reduces the dimension of the prediction space for the classifiers. The seen and the unseen experts of GatingAE predict a class out of $|\mathcal{Y}^S|$ or $|\mathcal{Y}^U|$ classes, respectively, instead of total $|\mathcal{Y}^S| + |\mathcal{Y}^U|$ number of classes as in the *No gating* model. The reduction of prediction space allows experts to focus on less number of classes for the classification, which lead to better accuracy performance. With these two advantages, GatingAE significantly improves the harmonic mean accuracy.

Advantage of using an independently trained expert (1-NN vs. Linear CLF) We compare the performance of GatingAEs using the 1-NN classifier and the linear classifier (Linear CLF) as seen experts. By comparing these two models, we emphasize the advantage of GatingAE that it can decompose the entire GZSL framework into three components, which are a gating model, a seen expert, and an unseen expert, and independently improve each expert. For instance, we can train a linear classifier as a seen expert independently from other two components using available visual training data. We show that the linear classifier can improve the seen accuracy without sacrificing unseen class accuracy. In Ta-

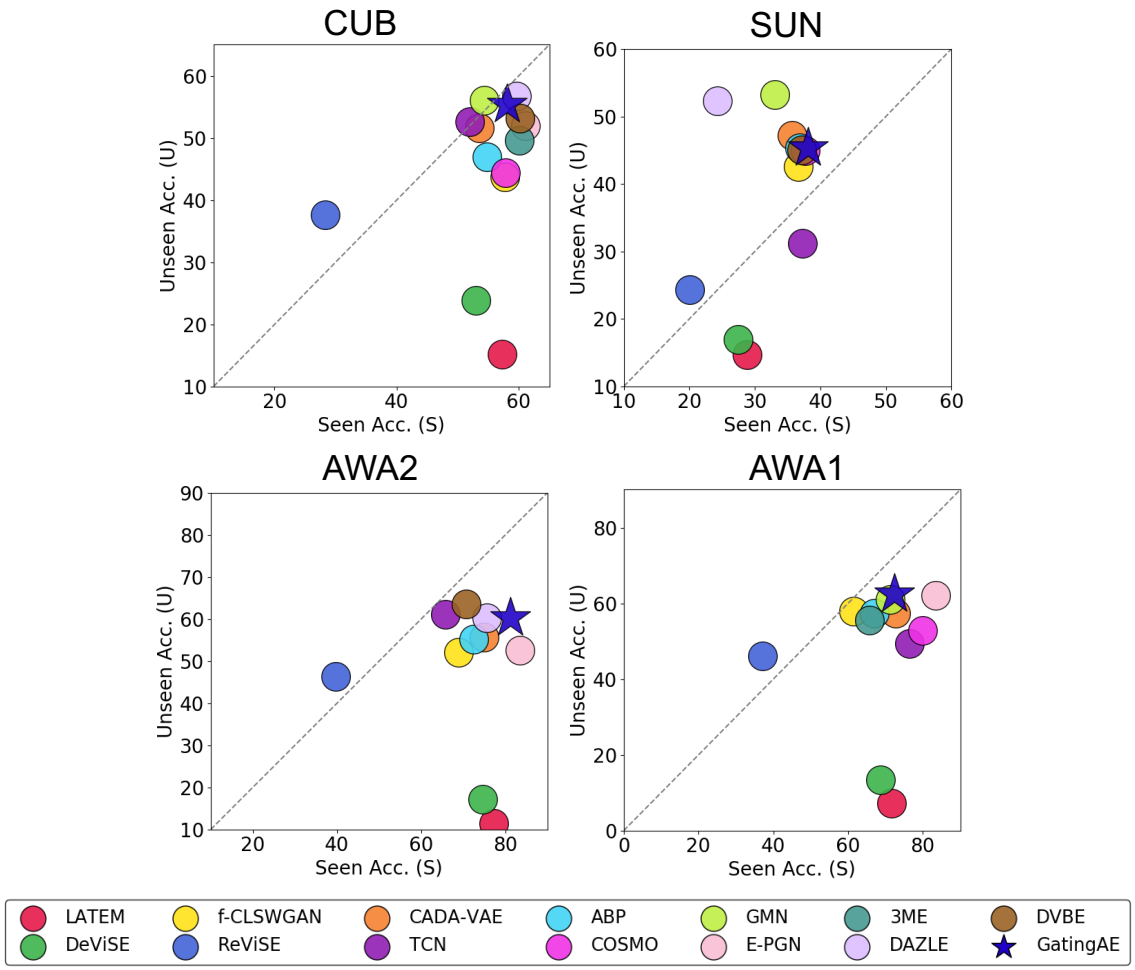


Figure 6.2: Scatter plot of seen and unseen accuracy for each state-of-the-art algorithm. For an ideal GZSL algorithm, the data point is expected to stay close the middle gray dotted line and the top right corner.

ble 6.1, GatingAE (r_{latent}) with the linear classifier achieves higher seen class accuracy by a margin of 3.6, 5.9, 2.1, and 1.9 than the GatingAE with the 1-NN classifier in CUB, SUN, AWA2, and AWA1, respectively. Also, GatingAE (r_{cross}) with the linear classifier outperforms that with the 1-NN classifier by at least 1.8 in terms of the seen class accuracy. In both cases, the unseen class accuracy is not compromised and the harmonic mean accuracy is improved in all four datasets. This shows that GatingAE enables to separately improve the experts of GZSL framework and the overall performance.

Complementary distance features for gating (GatingAE (r_{latent}, r_{cross}) vs. Gatin-

Table 6.3: State-of-the-art comparison in CUB, SUN, AWA2, and AWA1 datasets. S: Seen class accuracy, U: Unseen class accuracy, H: Harmonic mean accuracy. Top 2 harmonic mean accuracies for each dataset are highlighed in bold.

Method	CUB			SUN			AWA2			AWA1		
	S	U	H	S	U	H	S	U	H	S	U	H
LATEM [71]	57.3	15.2	24.0	28.8	14.7	19.5	77.3	11.5	20.0	71.7	7.3	13.3
DeViSE [66]	53.0	23.8	32.8	27.4	16.9	20.9	74.7	17.1	27.8	68.7	13.4	22.4
f-CLSWGAN [75]	57.7	43.7	49.7	36.6	42.6	39.4	68.9	52.1	59.4	61.4	57.9	59.6
ReViSE [67]	28.3	37.6	32.3	20.1	24.3	22.0	39.7	46.4	42.8	37.1	46.1	41.1
CADA-VAE [84]	53.5	51.6	52.4	35.7	47.2	40.6	75.0	55.8	63.9	72.8	57.3	64.1
TCN [73]	52.0	52.6	52.3	37.3	31.2	34.0	65.8	61.2	63.4	76.5	49.4	60.0
ABP [81]	54.8	47.0	50.6	36.8	45.3	40.6	72.6	55.3	62.6	67.1	57.3	61.8
COSMO [87]	57.8	44.4	50.2	37.7	44.9	41.0	-	-	-	80.0	52.8	63.6
GMN [78]	54.3	56.1	55.2	33.0	53.2	40.7	-	-	-	71.3	61.1	65.8
E-PGN [80]	61.1	52.0	56.2	-	-	-	83.5	52.6	64.6	83.4	62.1	**71.2**
3ME [120]	60.1	49.6	54.3	-	-	-	-	-	-	65.7	55.5	60.2
DAZLE [74]	59.6	56.7	**58.1**	24.3	52.3	33.2	75.7	60.3	67.1	-	-	-
DVBE* [88]	60.2	53.2	56.5	37.2	45.0	40.7	70.8	63.6	67.0	-	-	-
GatingAE	58.1	54.9	56.4	38.1	45.4	**41.4**	81.3	57.3	**67.2**	72.8	59.7	65.6
GatingAE + f-CLSWGAN	58.1	55.4	**56.7**	38.1	45.3	**41.4**	81.3	60.3	**69.3**	72.3	62.5	**67.2**

gAE (r_{all})) We compare GatingAE (r_{latent}) and GatingAE (r_{cross}) with GatingAE (r_{all}) and show that the combination of the distance features from the latent space and the cross-reconstruction space can further improve the GZSL performance. In particular, we compare GatingAEs using the linear classifiers as seen experts because they achieve better performance than GatingAEs using the 1-NN classifiers. In Table 6.1, GatingAE (r_{all}) consistently achieves higher harmonic mean accuracy than GatingAE (r_{latent}) and GatingAE (r_{cross}) across all the datasets except that GatingAE (r_{all}) achieves the same harmonic mean accuracy as GatingAE (r_{cross}) in CUB.

We believe the better performance of GatingAE (r_{all}) is resulted from the complementary distance features obtained in the latent space and the cross-reconstruction space. GatingAE (r_{latent}) shows higher seen class accuracy than GatingAE (r_{cross}) while GatingAE (r_{cross}) shows higher unseen class accuracy than GatingAE (r_{latent}) in Table 6.1. Since both models are based on the same seen and unseen experts, this comparison of the seen and the unseen accuracy shows that gating based on r_{latent} detects seen class samples well while gating based on r_{cross} detects unseen class samples better. Also, considering that the distance features from the latent space is lower dimensional than those from the cross-reconstruction space, the the distance features from different spaces perform gating at different levels of data abstraction. Hence, both distance features possess their own advantages for characterizing unseen class samples and can be used as complementary features. By combining both features for r_{all}, GatingAE (r_{all}) utilizes the advantages of each feature and achieves higher harmonic mean accuracy than both GatingAE (r_{latent}) and GatingAE (r_{cross}).

6.3.3 Comparison With State-of-the-art Algorithms

We compare GatingAE with 13 state-of-the-art GZSL algorithms and report the performance in Table 6.3. A hyphen (-) indicates that the authors of the algorithm have not validated their method in the corresponding dataset. For fair comparison with DVBE, we

use their reported performance without finetuning the backbone architecture of ResNet-101 for visual feature extraction. Excluding GatingAE + f-CLSWGAN, the base GatingAE achieves the best harmonic mean accuracy in SUN and AWA2, and the third highest harmonic mean accuracy in CUB and AWA1. Although GatingAE does not achieves the best performance in CUB and AWA1, GatingAE performs more robustly across datasets compared to other algorithms. For instance, DAZLE achieves the highest harmonic mean accuracy in CUB but its harmonic mean accuracy in SUN is 9^{th} highest out of 14 algorithms. Also, although E-PGN achieves the highest harmonic mean accuracy in AWA1, its harmonic mean accuracies in CUB and AWA2 are both 4^{th} highest out of 14. GatingAE achieves the highest average rank of 2 over all four datasets in terms of the harmonic mean accuracy. In comparison with the state-of-the-art soft-gating model COSMO, GatingAE which is based on the hard-gating achieves better performance in all datasets. Since the soft-gating model predicts a class using the combination of seen and unseen class prediction scores, the bias toward seen classes still affects the classification of unseen classes. However, GatingAE completely separates the classification of seen and unseen classes and mitigates the effect of the bias in unseen class classification.

We visualize the seen and the unseen accuracies of all the state-of-the-art methods in Fig Figure 6.2 to analyze the balance between the seen and the unseen accuracies. In particular, the x-axis and the y-axis in each scatter plot indicate the seen accuracy and the unseen accuracy of each method, respectively. Also, the gray dotted line in the middle indicates the same seen and unseen accuracies. An ideal GZSL method should achieve high accuracy for both seen and unseen classes and should not be biased toward either seen classes or unseen classes. Therefore, the accuracy of the ideal method is expected to be plotted close to the top right corner while staying close the dotted gray line. In CUB, GatingAE is one of the most closest methods to the dotted gray line and the top right corner. While DAZLE and GMN are located close to GatingAE in CUB, they are biased toward the unseen class accuracy and located far away from the center line in SUN. Although there

are several methods staying close to the center line than GatingAE in AWA2, GatingAE still achieves the highest harmonic mean acccuracy in AWA2. In AWA1, GatingAE shows comparable performance to E-PGN and GMN while being located close to the center dotted line. This shows that GatingAE achieves generalized high accuracy performance for both seen and unseen classes across all four datasets.

We also show that GatingAE can be easily combined with other state-of-the art methods to further improve the performance. Since each expert can be independently improved in GatingAE, the state-of-the-art methods can be simply utilized as a seen or an unseen expert. Furthermore, GatingAE can benefit from the state-of-the-art methods based on generative models, although the state-of-the-art methods do not achieve better GZSL performance than GatingAE. As a case study, we use f-CLSWGAN which is one of the earliest GZSL methods based on a WGAN [76]. f-CLSWGAN generates unseen visual features to tackle the problem of GZSL. We use these generated unseen visual features from f-CLSWGAN to finetune and improve the unseen expert independently from the seen expert. We report the performance of GatingAE + f-CLSWGAN in Table 6.3. The base GatingAE significantly outperforms f-CLSWGAN by a margin of 6.7, 2.0, 7.8, and 6.0 in CUB, SUN, AWA2, and AWA1, respectively. However, GatingAE still benefits from f-CLSWGAN and GatingAE + f-CLSWGAN achieves higher harmonic mean accuracy than individual GatingAE and f-CLSWGAN. Since we only finetune the unseen expert, GatingAE + f-CLSWGAN improves the unseen class accuracy over GatingAE while keeping the seen accuracy intact. Also, GatingAE + f-CLSWGAN achieves the best performance in SUN and AWA2, and the second best performance in CUB and AWA1 in terms of the harmonic mean accuracy. Although we only show one case study of using f-CLSWGAN, the same finetuning approach can be utilized with other GZSL algorithms based on generative models such as [77, 78, 82].

6.3.4 Ablation Study

Gating performance comparison with COSMO We compare the gating performance of GatingAE with the state-of-the-art gating method COSMO in Table 6.2. In particular, the authors of COSMO split the validation set into a Gating-Train set and a Gating-Val set and report the gating performance in the Gating-Val set. Following the same protocol, we also train the two-stream autoencoder in the original training set, tune the hyperparameters in the Gating-Train set, and finally report the gating performance in the Gating-Val set. We compare GatingAEs based on r_{latent}, r_{cross}, r_{all} with two gating models proposed in COSMO, which are MAX-SOFTMAX-3 and GB-GATING-3. The gating performance is validated in terms of the harmonic mean accuracy, AUC, and FPR. We note that GatingAE achieves higher harmonic mean accuracy than COSMO in all the test sets of CUB, SUN, and AWA1 as shown in Table 6.3.

GatingAEs significantly outperform MAX-SOFTMAX-3 and GB-GATING-3 in terms of all evaluation metrics in CUB. This further supports the significant performance gap between GatingAE and COSMO in the test set of CUB shown in Table 6.3. In SUN, while GatingAE (r_{all}) achieves slightly lower harmonic mean accuracy in the Gating-Val set compared to CB-GATING-3, it achieves better detection performance with higher AUC and lower FPR. In AWA1, GatingAE (r_{latent}) and GatingAE (r_{all}) achieve significantly higher harmonic mean accuracy while achieving lower AUC and higher FPR than CB-GATING-3 in the GZSL-val set. As shown in Table 6.3, GatingAE (r_{all}) outperforms COSMO by a large margin of 2.0 harmonic mean accuracy in the test set of AWA1. Considering that the GZSL-Val set is around three time smaller than the test set of AWA1, we argue that GatingAE (r_{all}) maintains its gating performance and learns better class discriminant representations in the relatively large-scale test set of AWA1.

Analysis on the gating performance from each distance feature We decompose the unseen class scores used in GatingAE to understand the contribution of each distance feature on gating. In particular, we report the AUC scores obtained by separately using the

Table 6.4: AUC performance obtained from using the distance in the latent space and the cross-reconstruction space as an unseen class score.

Unseen Class Score	CUB	SUN	AWA2	AWA1
d^S_{latent}	0.511	0.546	0.686	0.650
$1/d^U_{latent}$	0.596	0.520	0.459	0.516
$d^S_{latent}/d^U_{latent}(r_{latent})$	**0.842**	**0.774**	**0.934**	**0.917**
d^S_{cross}	0.496	0.550	0.725	0.697
$1/d^U_{cross}$	0.574	0.500	0.421	0.427
$d^S_{cross}/d^U_{cross}(r_{cross})$	**0.808**	**0.769**	**0.933**	**0.907**
$r_{cross} + \beta r_{latent}(r_{all})$	**0.841**	**0.783**	**0.940**	**0.918**

Table 6.5: Comparison of the number of model parameters between GatingAE and other generative model-based GZSL algorithms.

Model	f-CLSWGAN [75]	CADA-VAE [84]	GatingAE
# of parameters	19,514,062	7,398,716	5,860,138

latent space distance features, d^S_{latent} and $1/d^U_{latent}$, and the cross-reconstruction space distance features, d^S_{cross} and $1/d^U_{cross}$, as unseen class scores. Also, we compare the AUC scores from individual distance features with those from r_{latent}, r_{cross}, and r_{all}, which are the combination of the distance features. This highlights that the distance features are complementary to each other for gating. We report the AUC scores obtained in the test set of CUB, SUN, AWA2, and AWA1 in Table 6.4. $d^S_{latent}/d^U_{latent}$ and d^S_{cross}/d^U_{cross} shows significant improvement of the AUC scores over d^S_{latent}, $1/d^U_{latent}$, d^S_{cross}, $1/d^U_{cross}$. This shows that the distance features from seen and unseen classes are combined to effectively classify whether the query is from seen classes or unseen classes. In addition, r_{all} shows higher AUC than r_{latent} and r_{cross} in SUN, AWA2, and AWA1. In CUB, r_{latent} performs marginally better than r_{all}. We argue that GatingAE combines all the complementary distance features from seen classes, unseen classes, latent space, and cross-reconstruction space to achieve accurate gating results, and consequently choose correct experts for tackling GZSL problems.

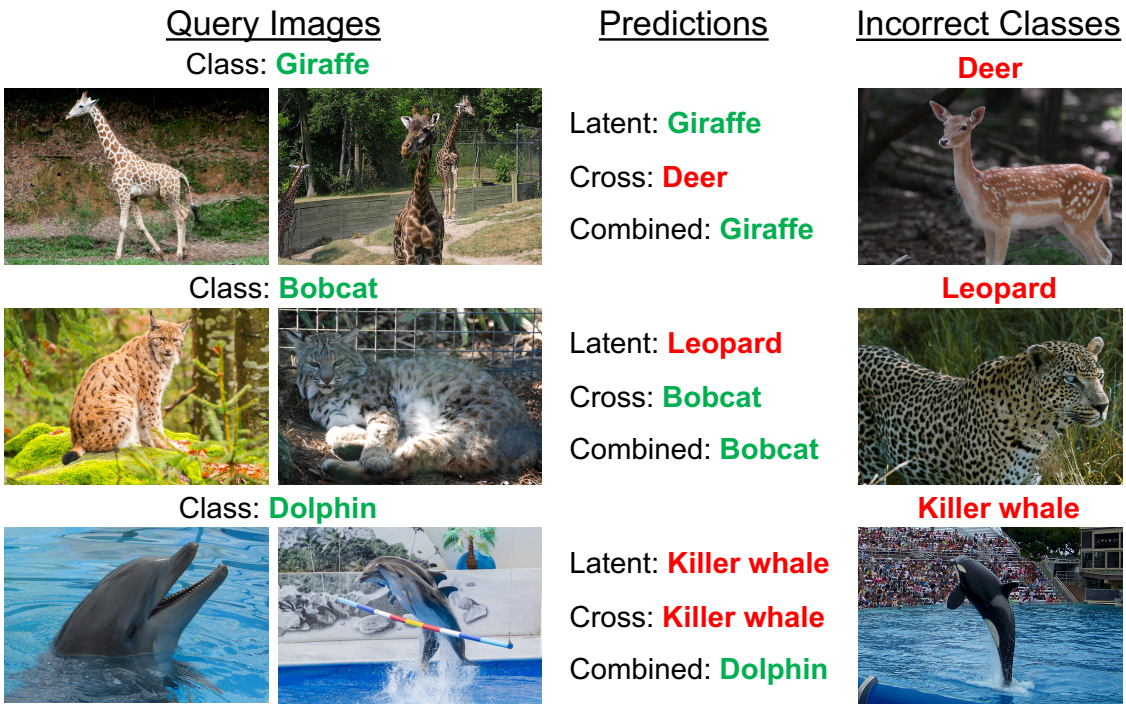


Figure 6.3: Qualitative analysis on the failure cases of GatingAEs using unseen class scores from different representation spaces in AWA2. Latent, Cross, and Combined refer to the class predictions of GatingAEs using r_{latent}, r_{cross}, and r_{all}, respectively.

Qualitative analysis on complementary distance features We perform qualitative analysis on the failure cases of GatingAEs using r_{latent}, r_{cross}, and r_{all} in Figure 6.3. In particular, unseen class query images are given to GatingAEs and we analyze cases where either GatingAE (r_{latent}) or GatingAE (r_{cross}) fails, and both of them fail in predicting correct classes. Through this analysis, we further highlight that the latent and the cross-reconstruction features capture unseen classes at different levels of data abstraction. The distance features from the low dimensional latent space focus more on abstracted global features while the cross-reconstruction features capture low-level local characteristics. In the first row of Figure 6.3, only GatingAE (r_{cross}) misclassfies the unseen class `Giraffe` into the seen class `Deer`. `Giraffe` and `Deer` shares local features of brown and white furs. However, they are clearly distinguished by the global features of the `Giraffe` such as a long neck and long legs. The unseen class score r_{latent} captures these global features that r_{cross} misses to predict the correct class. In the second row, `Bobcat` is misclassified

as `Leopard` by GatingAE (r_{latent}) while being correctly classified by GatingAE (r_{cross}). Since both classes are in the cat family, they are differentiated only by low-level local features such as sharpness of the ears and body patterns. We believe these local features are better captured by r_{cross} than r_{latent}. Finally, in the last row, we show two examples of the `Dolphin` class where both GatingAE (r_{latent}) and GatingAE (r_{cross}) misclassify them into the seen class `Killer whale` while GatingAE (r_{all}) correctly predicts the unseen class. `Dolphin` shares most of the characteristic features with and `Killer whale`, which makes unseen class detection challenging. However, we incorporate both local and global features from r_{latent} and r_{cross} in GatingAE (r_{all}) and achieve the correct prediction. For all the query images given in Figure 6.3, GatingAE (r_{all}) predicts the correct classes when one or both of GatingAE (r_{latent}) and GatingAE (r_{cross}) fail. This show that GatingAE (r_{all}) effectively combines the advantages of each feature abstracted at different levels.

Computational efficiency of GatingAE GatingAE is computationally efficient because of its compact two-stream linear autoencoder and the unified framework for the gating model and the unseen expert. To highlight the computational efficiency of GatingAE, we compare the number of parameters required to be trained for GatingAE with f-CLSWGAN and CADA-VAE in Table 6.5. Several state-of-the-art methods such as [77, 78] are developed on top of f-CLSWGAN. Also, CADA-VAE uses a two-stream VAE which is the closest architecture to our two-stream linear autoencoder. By comparing with these two models which are based on the simple generative models, we emphasize that GatingAE is even simpler while achieving the state-of-the-art performance. As shown in Table 6.5, f-CLSWGAN and CADA-VAE require around 3.3 times and 1.3 times more parameters than GatingAE, respectively. CADA-VAE uses the same number of layers and the same dimension for the latent space as GatingAE. However, CADA-VAE has to learn more parameters for a latent constraint and a classifiers for $|\mathcal{Y}^S| + |\mathcal{Y}^U|$ classes while GatingAE only needs to train a classifier for $|\mathcal{Y}^S|$ number of seen classes. In addition, the state-of-the art soft-gating model COSMO uses f-CLSWGAN as an unseen expert. Hence,

COSMO requires to train more than 19 million parameters of f-CLSWGAN. However, GatingAE uses the 1-NN classifier which does not need to train any additional parameters as an unseen expert. Therefore, GatingAE uses significantly less computational resources while outperforming these state-of-the-art methods.

6.4 Summary

We propose a GZSL algorithm, GatingAE, which utilizes the activation-based representations learned with auxiliary information to prevent biased prediction and achieve high accuracy performance for both seen and unseen classes data. In particular, we utilize distance features obtained from the latent space and the cross-reconstruction space of the two-stream autoencoder for gating. Based on the gating results, either the seen or the unseen class expert is chosen to perform the target task. We thoroughly validate the gating performance and the overall GZSL performance in the application of image recognition. GatingAE achieves the state-of-art performance in four benchmark image recognition datasets. Also, several advantages of GatingAE such as complementary distance features for gating, using independently trained experts, and computational efficiency are highlighted through baseline experiments and ablation studies. The characterization of the bias presented in the training data and the utilization of the bias information to calibrate the prediction for learning with limited data remains as future work.

CHAPTER 7
CONCLUSION

7.1 Contributions

We analyze different types of representations to successfully generalize to OOD. In particular, we investigate the capability of representations for differentiating ID from OOD. Based on the broad literature survey, we establish a new categorization of representations based on the information flow that the representations are obtained from. From the forward propagation, the activation-based representations are obtained. The gradient-based representations are generated by backpropagating the information through the network. Finally, both visual and attribute data as auxiliary information are forward propagated to obtain aligned representations for both data. We rigorously analyze their advantages and limitations for characterizing OOD.

First of all, we propose using backpropagated gradients as representations to characterize OOD. We extract the gradients from an autoencoder and use them as features to train a classifier for OOD detection. We conducted comprehensive baseline experiments to compare the OOD detection performance of standard activation-based representations and our proposed gradient-based representations. We show that the classifiers trained using the gradients as features outperform those trained using common activation-based features in OOD class and condition detection. These results validate the effectiveness of gradients as representations to capture OOD and lead us to develop more sophisticated representation learning techniques using gradients.

We develop a regularization technique for the gradient-based representations. To enhance the OOD detection capability of gradient-based representations, we use Fisher kernel to theoretically interpret the gradients. Based on the theoretical motivation, we propose

using a directional constraint for gradients. In particular, we constrain the direction of gradients from ID to be similar and measure the deviation of gradients from the directional constraint to detect OOD. From thorough baseline analysis, we show the effectiveness of the constrained gradient-based representations for OOD detection in comparison with the activation-based representations. Also, the proposed OOD detection algorithm, GradCon in benchmarking image recognition datasets. In addition, we show that GradCon requires significantly less number of model parameters to highlight its computational efficiency.

We thoroughly examine the advantages and the limitations of activation- and gradient-based representations for OOD characterization. During learning the gradient-based representations, we impose a directional constraint using two reference directions which are tangential and orthogonal directions with respect to the manifold. The tangential gradients represent ID and the orthogonal gradients represent OOD. While the gradient-based representations show better OOD performance than activation-based representations, the target tasks such as image classification cannot be directly performed using only gradient information. To complement the limitation, we propose learning aligned activation-based representations for visual and attribute data. The attribute representation can be utilized as another reference and we measure the distance from the visual representations to ID and OOD attribute representations for OOD characterization. From our controlled experiments, we validate that the activation-based representations learned with auxiliary information outperform both activation-based and gradient-based representations in terms of OOD detection performance.

Finally, we investigate using aligned representations with visual and attribute data to overcome one of the biggest challenges in generalization to OOD, which is the bias calibration for neural networks. In particular, we propose a GatingAE which utilizes the activation-based representations learned with auxiliary information to perform OOD detection and generalization. The gating model first detects whether the test image is from ID or OOD. Based on the gating results, either the ID seen class expert or the OOD unseen

class expert is chosen to perform the target task. We validate the gating performance and the overall generalization performance in OOD using the framework of generalized zero-shot learning. GatingAE achieves the state-of-art performance in four benchmark image recognition datasets significantly less number of training parameters in comparison with other state-of-the-art methods. The superior performance of GatingAE highlights the importance of learning effective representations that can distinguish OOD from ID and calibrating the biased prediction using OOD characterization.

7.2 Prospective Research Directions

Representation learning techniques discussed in this dissertation open up several promising research directions. First of all, the formulation of gradient constraint can be further improved to better characterize OOD. The developed gradient constraint simply aligns gradients by measuring the cosine similarity between the gradient from the current step and the average training gradient. The average training gradient is computed by taking an average of all the gradients from previous steps of training. Since all the previous gradients are taken into account, the average gradient represents the global trend of training gradients while not effectively characterizing local gradient changes. In addition, considering that the gradient changes drastically in the beginning of training, the gradients from a first few steps possibly become dominant in the average gradient calculation. We expect more sophisticated formulation of reference gradient can further enhance OOD detection performance.

Gradient-based representations from supervised networks also remains as a future research direction. We extract gradients from an unsupervised learning framework to obtain gradient-based representations. Since the annotation is not required to compute the loss in the unsupervised learning framework, the gradients can be easily obtained. However, for supervised networks, we still need to investigate the generation of gradients that can be useful as representations. Since the supervised network is trained with annotations which

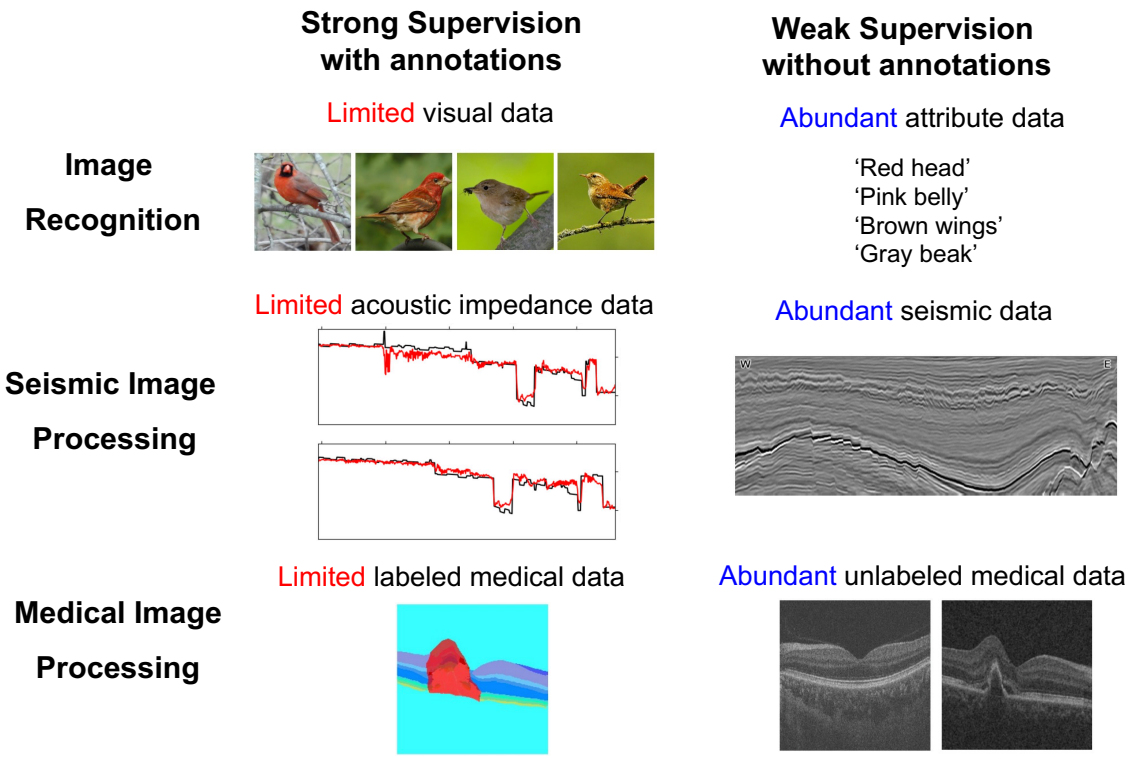


Figure 7.1: Illustration of applications which utilize limited annotated data and abundant unannotated data for training.

we do not have access to during testing, we need pseudo-labels to generate gradients. Designing the pseudo-labels is a significantly important step to obtain representations that can characterize OOD. The authors in [63] propose the concept of confounding label to generate gradients in the supervised learning framework. The gradients generated by the confounding label shows promising results in OOD detection. We believe more follow up works on gradient-based representations from the supervise networks will broaden the applicability of the representations.

Recent studies show promising research directions of multimodal learning for generalization to OOD. In `GatingAE`, we use visual and attribute data for multimodal training. The attribute information are utilized to transfer knowledge learned from ID visual data to OOD visual data. The authors in [121] recently scale up the multimodal training and show that models trained to align a large-scale multimodal data achieve surprisingly powerful zero-shot classification performance. They use 400 million image-text pairs crawled from

the internet to train a two-stream encoder models. The image encoder and the text encoder are trained to minimize a contrastive loss which enforces to align positive pairs of image-text data and push away negative pairs. While this approach achieves competitive zero-shot transfer performance compared to fully supervised models on over 30 different computer vision datasets, the authors state that the model still faces challenges in generalizing to specific OODs. Considering the importance and the challenges of generalization to OOD for real-world application, we believe OOD characterization using large-scale pretrained multimodal models is a significantly important research problem to achieve successful generalization to OOD.

Finally, bias calibration can be explored for applications which require learning with limited data. We validate bias calibration capability of `GatingAE` in the application of image recognition. However, there exist several other applications which require bias calibration particularly when only limited annotated data is available during training. For instance, data which requires professional knowledge for annotation such as medical images or seismic images is limited in terms of available annotated data. Recent works [122, 122] tackle vision problems with limited data using semi-supervised or weakly supervised learning frameworks. As shown in Figure 7.1, limited annotated data which provides strong supervision to the neural networks are available in seismic and medical image processing. Since this limited data is not enough to train neural networks, they additionally use abundant unannotated data such as images of seismic volumes or medical data. Although a small number of annotated data is available, strong supervision from annotations often makes the model easily overfit only to the annotated data. Therefore, bias calibration techniques can be further investigated to correct the bias and more effectively use unlabeled data to enhance the generalization capability of the models.

REFERENCES

[1] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[2] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning*, PMLR, 2017, pp. 1321–1330.

[3] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, p. 533, 1986.

[4] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.

[5] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, p. 788, 1999.

[6] A. S. Lewis and G. Knowles, "Image compression using the 2-d wavelet transform," *IEEE Transactions on image Processing*, vol. 1, no. 2, pp. 244–250, 1992.

[7] J.-L. Starck, E. J. Candès, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE Transactions on image processing*, vol. 11, no. 6, pp. 670–684, 2002.

[8] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[9] D. Temel and G. AlRegib, "Csv: Image quality assessment based on color, structure, and visual system," *Signal Processing: Image Communication*, vol. 48, pp. 92–103, 2016.

[10] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, *et al.*, "Image quality assessment: From error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[11] M. A. Aabed, G. Kwon, and G. AlRegib, "Power of tempospatially unified spectral density for perceptual video quality assessment," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2017, pp. 1476–1481.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[14] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[15] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT Press, 2016, vol. 1.

[16] A. Ng, "Sparse autoencoder," *CS294A Lecture notes*, vol. 72, no. 2011, pp. 1–19, 2011.

[17] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[19] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[20] X. J. Zhu, "Semi-supervised learning literature survey," 2005.

[21] I. Triguero, S. Garcia, and F. Herrera, "Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study," *Knowledge and Information systems*, vol. 42, no. 2, pp. 245–284, 2015.

[22] Z.-H. Zhou and M. Li, "Semi-supervised learning by disagreement," *Knowledge and Information Systems*, vol. 24, no. 3, pp. 415–439, 2010.

[23] A. Rasmus, H. Valpola, M. Honkala, M. Berglund, and T. Raiko, "Semi-supervised learning with ladder networks," *arXiv preprint arXiv:1507.02672*, 2015.

[24] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *arXiv preprint arXiv:1703.01780*, 2017.

[25] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, "Using self-supervised learning can improve model robustness and uncertainty," *arXiv preprint arXiv:1906.12340*, 2019.

[26] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *European conference on computer vision*, Springer, 2016, pp. 69–84.

[27] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European conference on computer vision*, Springer, 2016, pp. 649–666.

[28] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2536–2544.

[29] I. Misra and L. v. d. Maaten, "Self-supervised learning of pretext-invariant representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6707–6717.

[30] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.

[31] J. Platt *et al.*, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[32] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[33] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," *arXiv preprint arXiv:1706.02690*, 2017.

[34] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," *arXiv preprint arXiv:1807.03888*, 2018.

[35] T. DeVries and G. W. Taylor, "Learning confidence for out-of-distribution detection in neural networks," *arXiv preprint arXiv:1802.04865*, 2018.

[36] Y.-C. Hsu, Y. Shen, H. Jin, and Z. Kira, "Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 951–10 960.

[37] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," *arXiv preprint arXiv:1812.04606*, 2018.

[38] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[39] D. M. Tax and R. P. Duin, "Support vector data description," *Machine Learning*, vol. 54, no. 1, pp. 45–66, 2004.

[40] L. Ruff, N. Görnitz, L. Deecke, S. A. Siddiqui, R. Vandermeulen, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International Conference on Machine Learning*, 2018, pp. 4390–4399.

[41] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, ACM, 2014, p. 4.

[42] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2017, pp. 665–674.

[43] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," *International Conference on Learning Representations*, 2018.

[44] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara, "Latent space autoregression for novelty detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 481–490.

[45] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3379–3388.

[46] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International Conference on Information Processing in Medical Imaging*, Springer, 2017, pp. 146–157.

[47] S. Pidhorskyi, R. Almohsen, and G. Doretto, "Generative probabilistic novelty detection with adversarial autoencoders," in *Advances in Neural Information Processing Systems*, 2018, pp. 6822–6833.

[48] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.

[49] K. Lee, H. Lee, K. Lee, and J. Shin, "Training confidence-calibrated classifiers for detecting out-of-distribution samples," *arXiv preprint arXiv:1711.09325*, 2017.

[50] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*, Springer, 2014, pp. 818–833.

[51] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[52] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 618–626.

[53] M. Prabhushankar, G. Kwon, D. Temel, and G. AlRegib, "Contrastive explanations in neural networks," in *2020 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2020, pp. 3289–3293.

[54] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[55] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[56] H. Drucker and Y. Le Cun, "Double backpropagation increasing generalization performance," in *IJCNN-91-Seattle International Joint Conference on Neural Networks*, IEEE, vol. 2, 1991, pp. 145–150.

[57] A. S. Ross and F. Doshi-Velez, "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[58] J. Sokolić, R. Giryes, G. Sapiro, and M. R. Rodrigues, "Robust large margin deep neural networks," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4265–4280, 2017.

[59] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2007, pp. 1–8.

[60] A. Achille, G. Paolini, and S. Soatto, "Where is the information in a deep neural network?" *arXiv preprint arXiv:1905.12213*, 2019.

[61] A. Achille, M. Lam, R. Tewari, A. Ravichandran, S. Maji, C. C. Fowlkes, S. Soatto, and P. Perona, "Task2vec: Task embedding for meta-learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6430–6439.

[62] G. Kwon, M. Prabhushankar, D. Temel, and G. AIRegib, "Distorted representation space characterization through backpropagated gradients," in *2019 26th IEEE International Conference on Image Processing (ICIP)*, IEEE, 2019.

[63] J. Lee and G. AlRegib, "Gradients as a measure of uncertainty in neural networks," in *2020 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2020, pp. 2416–2420.

[64] F. Mu, Y. Liang, and Y. Li, "Gradients as features for deep representation learning," in *International Conference on Learning Representations*, 2020.

[65] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2009, pp. 951–958.

[66] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, "Devise: A deep visual-semantic embedding model," in *Advances in neural information processing systems*, 2013, pp. 2121–2129.

[67] Y.-H. Hubert Tsai, L.-K. Huang, and R. Salakhutdinov, "Learning robust visual-semantic embeddings," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3571–3580.

[68] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha, "Synthesized classifiers for zero-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5327–5336.

[69] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean, "Zero-shot learning by convex combination of semantic embeddings," *arXiv preprint arXiv:1312.5650*, 2013.

[70] Z. Zhang and V. Saligrama, "Zero-shot learning via joint latent similarity embedding," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 6034–6042.

[71] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele, "Latent embeddings for zero-shot classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 69–77.

[72] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *International Conference on Machine Learning*, 2015, pp. 2152–2161.

[73] H. Jiang, R. Wang, S. Shan, and X. Chen, "Transferable contrastive network for generalized zero-shot learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9765–9774.

[74] D. Huynh and E. Elhamifar, "Fine-grained generalized zero-shot learning via dense attribute-based attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4483–4493.

[75] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5542–5551.

[76] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[77] R. Felix, V. B. Kumar, I. Reid, and G. Carneiro, "Multi-modal cycle-consistent generalized zero-shot learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 21–37.

[78] M. B. Sariyildiz and R. G. Cinbis, "Gradient matching generative networks for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2168–2178.

[79] Y. Long, L. Liu, F. Shen, L. Shao, and X. Li, "Zero-shot learning using synthesised unseen visual data with diffusion regularisation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 10, pp. 2498–2512, 2017.

[80] Y. Yu, Z. Ji, J. Han, and Z. Zhang, "Episode-based prototype generating network for zero-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 035–14 044.

[81] Y. Zhu, J. Xie, B. Liu, and A. Elgammal, "Learning feature-to-feature translator by alternating back-propagation for generative zero-shot learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9844–9854.

[82] A. Mishra, S. Krishna Reddy, A. Mittal, and H. A. Murthy, "A generative model for zero shot learning using conditional variational autoencoders," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2188–2196.

[83] V. Kumar Verma, G. Arora, A. Mishra, and P. Rai, "Generalized zero-shot learning via synthesized examples," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4281–4289.

[84] E. Schonfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata, "Generalized zero- and few-shot learning via aligned variational autoencoders," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8247–8255.

[85] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-shot learning through cross-modal transfer," *Advances in neural information processing systems*, vol. 26, pp. 935–943, 2013.

[86] H. Zhang and P. Koniusz, "Model selection for generalized zero-shot learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[87] Y. Atzmon and G. Chechik, "Adaptive confidence smoothing for generalized zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 671–11 680.

[88] S. Min, H. Yao, H. Xie, C. Wang, Z.-J. Zha, and Y. Zhang, "Domain-aware visual bias eliminating for generalized zero-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 664–12 673.

[89] S. Liu, M. Long, J. Wang, and M. I. Jordan, "Generalized zero-shot learning with deep calibration network," *Advances in Neural Information Processing Systems*, vol. 31, pp. 2005–2015, 2018.

[90] W.-L. Chao, S. Changpinyo, B. Gong, and F. Sha, "An empirical study and analysis of generalized zero-shot learning for object recognition in the wild," in *European Conference on Computer Vision*, Springer, 2016, pp. 52–68.

[91] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[92] R. Timofte, K. Zimmermann, and L. Van Gool, "Multi-view traffic sign detection, recognition, and 3d localisation," *Machine vision and applications*, vol. 25, no. 3, pp. 633–647, 2014.

[93] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The german traffic sign recognition benchmark: A multi-class classification competition," in *The 2011 international joint conference on neural networks*, IEEE, 2011, pp. 1453–1460.

[94] ——, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural networks*, vol. 32, pp. 323–332, 2012.

[95] D. Temel, M. Chen, and G. AlRegib, "Traffic sign detection under challenging conditions: A deeper look into performance variations and spectral characteristics," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2019.

[96] D. Temel and G. AlRegib, "Traffic signs in the wild: Highlights from the ieee video and image processing cup 2017 student competition [sp competitions]," *IEEE Sig. Proc. Mag.*, vol. 35, no. 2, pp. 154–161, 2018.

[97] D. Temel, T. Alshawi, M.-H. Chen, and G. AlRegib, "Challenging environments for traffic sign detection: Reliability assessment under inclement conditions," *arXiv preprint arXiv:1902.06857*, 2019.

[98] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[99] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.

[100] D. Temel, G. Kwon, M. Prabhushankar, and G. AlRegib, "Cure-tsr: Challenging unreal and real environments for traffic sign recognition," *arXiv preprint arXiv:1712.02463*, 2017.

[101] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Advances in Neural Information Processing Systems*, 1999, pp. 487–493.

[102] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *International Journal of Computer Vision*, vol. 105, no. 3, pp. 222–245, 2013.

[103] X. Peng, C. Zou, Y. Qiao, and Q. Peng, "Action recognition with stacked fisher vectors," in *European Conference on Computer Vision*, Springer, 2014, pp. 581–595.

[104] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[105] P. Perera, R. Nallapati, and B. Xiang, "Ocgan: One-class novelty detection using gans with constrained latent representations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2898–2906.

[106] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[107] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[108] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, vol. 2, 2006, pp. 1735–1742.

[109] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, *et al.*, "Conditional image generation with pixelcnn decoders," in *Advances in Neural Information Processing Systems*, 2016, pp. 4790–4798.

[110] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International Conference on Information Processing in Medical Imaging*, Springer, 2017, pp. 146–157.

[111] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International Conference on Machine Learning*, 2018, pp. 4393–4402.

[112] P. Luc, N. Neverova, C. Couprie, J. Verbeek, and Y. LeCun, "Predicting deeper into the future of semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 648–657.

[113] Q. Zhao, L. Zong, X. Zhang, Y. Li, and X. Tang, "A multimodal clustering framework with cross reconstruction autoencoders," *IEEE Access*, 2020.

[114] Y. Yu, Z. Ji, J. Han, and Z. Zhang, "Episode-based prototype generating network for zero-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 035–14 044.

[115] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.

[116] G. Patterson and J. Hays, "Sun attribute database: Discovering, annotating, and recognizing scene attributes," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 2751–2758.

[117] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 9, pp. 2251–2265, 2018.

[118] S. Reed, Z. Akata, H. Lee, and B. Schiele, "Learning deep representations of fine-grained visual descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 49–58.

[119] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[120] R. Felix, M. Sasdelli, I. Reid, and G. Carneiro, "Multi-modal ensemble classification for generalized zero shot learning," *arXiv preprint arXiv:1901.04623*, 2019.

[121] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," *arXiv preprint arXiv:2103.00020*, 2021.

[122] Y. Alaudah, M. Alfarraj, and G. AlRegib, "Structure label prediction using similarity-based retrieval and weakly supervised label mapping: Geophysics, 84," *V67–V79*, 2019.

VITA

Gukyeong Kwon received his M.S. degree and Ph.D. degree from the School of Electrical and Computer Engineering (ECE) at Georgia Institute of Technology in 2018 and in 2021, respectively. He is a co-recipient of the Finalist of the World's First 10K Best Paper Award at the IEEE International Conference on Multimedia and Expo in 2017, the Best Paper Award at the IEEE International Conference on Image Processing in 2019, and the Top Viewed Special Session Paper Award at the IEEE International Conference on Image Processing in 2020. His research has primarily focused on the robustness of machine learning, multimodal representation learning, and learning with limited data.